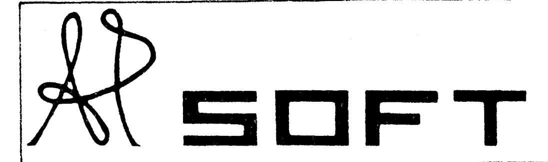
# La revue des SHARPENTIERS

SHARP ENSEIGNEMENT

N° 11 - BIMESTRIEL - MARS 85 - 25 F.

LA REVUE DES UTILISATEURS DE MICRO-ORDINATEURS ET POCKET-COMPUTERS SHARP




# UN NOUVEAU SOUFFLE POUR VOTRE POCKET.

## LOGICIELS POUR VOTRE PC 1500

| | |
|---|---|
| **SOFT CALC** | : tableur professionnel avec graphiques.Utilisable sur imprimante parallèle . |
| **SOFT GRAPH** | : graphiques de gestion et scientifiques , neuf types de graphiques . |
| **SOFT STAT** | : traitement statistique des données avec représentation graphique . |
| **SOFT TEXT** | : traitement de textes sur imprimante CE 150 et imprimante paralléle . |
| **SOFT DRAW** | : représentation graphique en trois dimensions , véritable mini CAO . |
| **SOFT FONCTION** | : tracé de courbes et de surfaces , merveilleux outil mathématique . |
| **SOFT EXTEND** | : extension de 70 fonctions au BASIC , une nouvelle puissance pour votre 1500. |
| **SOFT SYSTEM** | : gestion d'une partie de la RAM en disquette , inclu un mini CALC et un éditeur. |
| **SOFT MIND** | : 3 jeux utilisant les grands principes de l'intelligence artificielle . |
| **GAME 1 et 2** | : des jeux video en langage machine , un graphisme époustouflant . |

## LOGICIELS POUR VOTRE PC 1350

| | |
|---|---|
| **FREE CALC** | : tableur professionnel , possibilités de graphiques à l'écran . |
| **FREE GRAPH** | : graphiques de gestion et scientifiques (nécéssite l'imprimante CE-515 P) . |
| **FREE STAT** | : traitement statistique des données , compatible CALC et GRAPH . |
| **FREE TEXT** | : traitement de textes sur CE 126 P et imprimante série par l'interface 130 T . |
| **FREE BASE** | : puissante gestion de fichiers avec possibilité de calcul . Compatible CALC . |
| **FREE DRAW** | : représentations graphiques en trois dimensions (sur l'imprimante CE -515 P) |
| **JEUX 1 et 2** | : ensembles de jeux vidéo , écrits pour la plupart en langage machine . |
| **JEUX 3** | : les grands classiques , OTHELLO , MORPION et PUISSANCE 4 . |
| **JEUX 4** | : jeux de dés et de cartes , découvrez chez vous les joies de LAS VEGAS . |
| **JEUX 5** | : jeux de reflexion , MIXALETTRE, HANOI, LOGISUITE, PENDU et PUZZLE . |

**Tous nos logiciels sont livrés sur cassette audio avec une notice explicative détaillée . IlS sont disponibles chez votre revendeur SHARP , au CLUB DES SHARPENTIERS , dans les FNACS , aux NOUVELLES GALERIES et à EUROMARCHE.**

**REVENDEURS**

**contactez-nous au 976 83 74**

**Pour tous renseignements , contactez :**

**AP SOFT** Mr Pascal ABRIVARD
1 bis rue du Printemps 78230 LE PECQ
Tel. : 16 (3) 976 83 74

Vous pouvez aussi obtenir nos logiciels par correspondance :

**Nom** ............................................ **Adresse** ............................................
............................................................................ **Téléphone** ..........................................

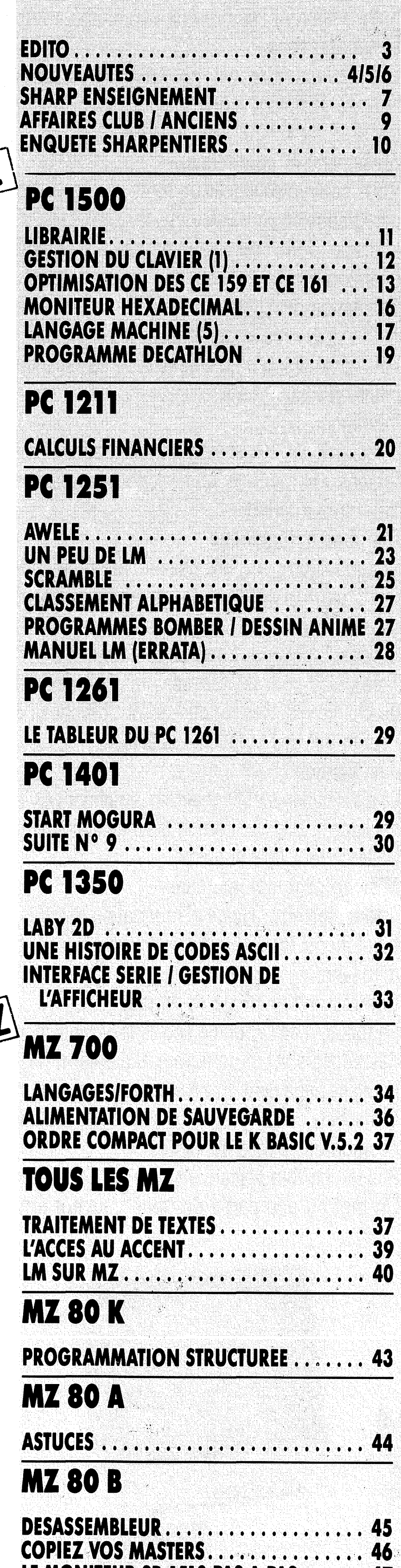
| | | |
|---|---|---|
| ☐ SOFT CALC ........ 200.00 FF | ☐ SOFT GRAPH ....... 150.00 FF | ☐ SOFT STAT .......... 150.00 FF |
| ☐ SOFT TEXT ........ 150.00 FF | ☐ SOFT DRAW ........ 150.00 FF | ☐ SOFT FONC ......... 150.00 FF |
| ☐ SOFT EXTEND ... 200.00 FF | ☐ SOFT SYSTEM ...... 200.00 FF | ☐ SOFT MIND ......... 80.00 FF |
| ☐ SOFT GAME 1.... 80.00 FF | ☐ SOFT GAME 2 ..... 80.00 FF | ☐ FRAIS DE PORT 15.00 FF |
| ☐ FREE CALC ......... 200.00 FF | ☐ FREE GRAPH ....... 150.00 FF | ☐ FREE STAT .......... 150.00 FF |
| ☐ FREE TEXT ......... 200.00 FF | ☐ FREE BASE ......... 200.00 FF | ☐ FREE DRAW ........ 150.00 FF |
| ☐ JEUX 1 ............... 80.00 FF | ☐ JEUX 2 ............... 80.00 FF | ☐ JEUX 3 ................ 80.00 FF |
| ☐ JEUX 4 ............... 80.00 FF | ☐ JEUX 5 ............... 80.00 FF | ☐ FRAIS DE PORT 15.00 FF |

**CI-JOINT** ........................... **Francs par** ☐ **CHEQUE** ☐ **CCP** ☐ **MANDAT**

**(Pour le 1350 contactez nous au préalable pour connaitre la liste des logiciels disponibles )**



## SOMMAIRE N° 11





## LE SHARPENTIER EDITO

Etudiants, étudiantes, professeurs, professeurs, le sigle de notre couverture est le vôtre. Vous le retrouverez souvent dans votre établissement ou à la vitrine de votre point de vente SHARP. Lisez la page ENSEIGNEMENT de ce numéro, réclamez votre chèque SHARP à votre prof' de maths ou de physique et profitez d'un prix exceptionnel sur l'achat d'une calculatrice scientifique ou d'un PC. Et, si cet achat vous a complètement ruiné, retournez voir votre prof', demandez-lui de vous prêter le dernier bulletin du Club ; il ne pourra rien vous refuser car, s'il participe à la CAMPAGNE ENSEIGNEMENT SHARP, nous l'avons inscrit gratuitement au Club pendant 1 an.

Justement, ce bulletin, comment le trouvez-vous ? Que pensez-vous du Club ? Aimeriez-vous qu'il vous apporte autre chose ? Notre ENQUETE SHARPENTIERS vous pose toutes ces questions ; soyez nombreux à y répondre. Nous tiendrons compte scrupuleusement, de toutes vos réponses et commentaires ; ils détermineront les points particuliers vers lesquels nous porterons nos efforts dans les mois à venir.

Une information pour conclure : ce bulletin vous sera expédié le 20 février 1985 ; l'élasticité des délais postaux concernant les envois en nombre fait qu'il vous parvient entre 3 et... 25 jours après son dépôt. Alors, un seul conseil : patience !

A bientôt.

S. BIZOIRRE

### CRITERIUM SHARPENTIER

Les règles en sont très simples mais peuvent rapporter gros aux SHARPENTIERS courageux :

1. Tout Sharpentier dont l'article, le programme ou l'astuce aura été publié dans le bulletin SHARPENTIER sera d'office inscrit au club, gratuitement, pour une nouvelle année.
2. A la parution de chaque bulletin, une main innocente tirera au sort l'un de ces auteurs qui gagnera ainsi un des plus récents produits SHARP.

Le tirage au sort des auteurs du bulletin n° 10 a été effectué le mercredi 3 janvier 1985. L'heureux gagnant est M. Bernard KOKANOSKI, auteur du SUPER BASIC pour MZ 700. Il gagne ainsi un PC 1260.

Pour le numéro 11, il y a un PC 1350 à gagner. A vos plumes...

### LE CLUB RECRUTE

- La micro-informatique vous passionne
- Vous avez des notions de secrétariat et de comptabilité
- Vous avez le goût des contacts
- Une ambiance sympathique n'est pas pour vous déplaire
- Vous êtes disponible à temps plein rapidement
- Vous habitez en région parisienne

**CONTACTEZ-NOUS VITE, NOUS AVONS DE GRANDS PROJETS POUR VOUS**

S. BIZOIRRE (833.93.44)


**ONT PARTICIPE A CE NUMERO**

**REDACTEUR EN CHEF**
**S. BIZOIRRE**
**REDACTEUR EN CHEF ADJOINT**
**L. BURELLER**

**PUBLICITE S. BIZOIRRE**
**834.93.44**

M. GIRONDOT, D. SATIN, D. SERGE, G. MALLET, M. RIVET, P. ABRIVARD, M. CHOUCROUN, J.-B. VIMOND, F. BOESSER, J.-F. VIGNAUD, G. NICOLAS, P. PIERSON, J. KOVACS, D. BEURRIER, M. LUCAS., B. KOKANOSKI, H. BENOIT, A. BERMOND, P. GAC, J.-M. SUPOR, J. MILLET, RENAUD

MAQUETTE ET FABRICATION IN-QUARTO. 19, RUE F.LEMAITRE 75020 PARIS. TEL. 636.73.04



Club des Sharpentiers 151/153, avenue Jean-Jaurès 93307 AUBERVILLIERS CEDEX Tél : 834.93.44




## LA RECETTE DU TRAITEMENT DE TEXTES IDEAL

1 / Munissez-vous d'une machine à écrire SHARP ZX400, ZX410 ou ZX330.
2 / Reliez-la à une unité centrale SHARP préalablement équipée d'un programme adéquat en mémoire morte et de 32K ou 48K de mémoire vive.
3 / Ajoutez, à votre convenance, 1 ou 2 unités de disquettes 5 pouces.
4 / Ornez le tout d'un moniteur vert SHARP 12 pouces.
Vous obtenez, ainsi, un système de visualisation et de mémorisation de tous vos textes tapés à la machine. Très simple d'utilisation, muni d'une mémoire protégée, il ne nécessite aucun chargement de programme lors de sa mise en route.

## L'ÉCRAN

L'écran affiche 20 lignes de 80 caractères pour visualiser le contenu de la mémoire exactement comme cela sera imprimé. Avec l'écran, la dactylographie devient plus agréable.

## LA MÉMOIRE CENTRALE

La version de base comprend une mémoire de 32 K qui peut être étendue à 48 K par un module d'extension de 16 K.
Cette mémoire se répartit entre : documents, constantes et imprimés.

## PROTECTION DE LA MÉMOIRE

Le programme, ainsi que le contenu des mémoires restent protégés en cas de panne de courant ou lorsque la machine est hors tension. La durée de protection est de trois mois.

## MACHINE A ÉCRIRE

Une simple pression sur M et votre machine à écrire retrouve son autonomie et ses propres fonctions.

## SUPPORT EXTERNE

La version de base est équipée d'une station de lecture. En fonction de vos besoins, une deuxième station peut être installée.

Disquette : 5 1/4" - simple face, double densité - 156 K.

## TROIS TYPES DE MÉMOIRES

**Mémoire de documents**
Cette mémoire permet de stocker des textes

LES NOUVEAUTÉS

divers. La gestion de cette mémoire s'opère automatiquement et il est aisé de retrouver un document dans la mémoire de travail car celle-ci se feuillette comme un classeur.
Les textes qui sont mémorisés sur disquette peuvent être appelés à tout moment.

**Mémoire de constantes**
Des textes répétitifs, tels que formules de politesse, formules chimiques ou autres, peuvent être stockés dans cette mémoire et être transférés en mémoire de document.
Chaque constante est identifiée pour être appelée et permettre de composer un texte à l'écran.

**Mémoire d'imprimés**
Cette mémoire permet de préparer différentes lignes format : marges, tabulations, longueur de l'imprimé, pas d'écriture, interligne, repère de positionnement (pour les formulaires). A chaque papier à en-tête ou formulaire utilisés correspondra un "imprimé".

## IMPRESSION

Tout document affiché à l'écran peut être imprimé. Il est possible de demander l'impression d'un texte de plusieurs pages à partir d'une page quelconque.

## IMPRESSION SIMULTANÉE

Plus de temps perdu en attendant la fin de l'impression. Il est possible, alors que votre machine à écrire imprime, de saisir un nouveau texte à l'écran.

## LANGUES ET CLAVIERS

La disquette programme a été conçue pour être utilisée en plusieurs langues. A vous de choisir la vôtre et de sélectionner le clavier correspondant à votre machine à écrire.

## FONCTIONS DE TRAITEMENT DE TEXTES

Cet ensemble permet l'accès à différentes fonctions de traitement de textes. La visualisation à l'écran rend le travail plus agréable.




**Suppression et insertion de caractères, mots ou paragraphes**
Elles peuvent être réalisées en cours ou après la saisie, la machine redisposant automatiquement et immédiatement le texte.

**Soulignement automatique**
Vous pouvez souligner un mot, une ligne, un paragraphe en cours ou après la saisie. La suppression d'un soulignement peut intervenir ultérieurement.

**Tabulation décimale**
Tout taquet simple peut être utilisé pour la fonction décimale. L'alignement peut se faire au point ou à la virgule.

**Interligne**
Il est possible, à tout moment, de modifier l'interligne. Un même document peut comporter différents interlignes.

**Centrage automatique**
Chaque ligne peut être centrée automatiquement entre marge droite et marge gauche, avant ou après la frappe.

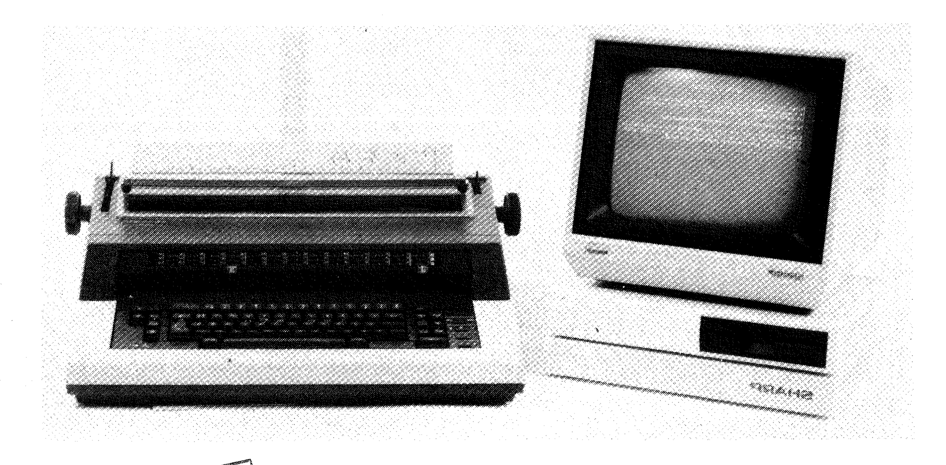
**Indices/Exposants/Caractères composés**
Très utiles pour les formules mathématiques ou chimiques. Ces fonctions sont affichées à l'écran par des symboles spéciaux.

**Justification**
La justification s'obtient automatiquement ligne par ligne ou bien sur l'ensemble d'un document. Celle-ci peut être annulée.

**Mise en page automatique**
A tout moment, la ligne d'information (marges, tabulations, longueur imprimé, etc.) peut être modifiée.

**Contrôle automatique du format**
En fonction de l'imprimé choisi, la composition des pages s'effectue automatiquement. Une nouvelle page peut être créée par vous-même, la machine opérant automatiquement la recomposition des pages suivantes.

**Recherche/Remplacement**
Un mot ou une partie de texte peuvent être



recherchés automatiquement en une fraction de seconde et être remplacés par d'autres.

**Marquage de textes**

En délimitant des parties de texte vous pouvez :

- Transférer un ou plusieurs paragraphes
- Copier un ou plusieurs paragraphes
- Annuler un ou plusieurs paragraphes
- Justifier un ou plusieurs paragraphes
- Mémoriser un ou plusieurs paragraphes sur disquette pour permettre un assemblage de textes.

**Défilement horizontal**

Pour obtenir, à l'écran, un affichage identique à l'impression de documents dont les marges sont supérieures à 80 caractères.

**Relocalisation**

Vous pouvez aller directement au début ou à la fin d'une ligne ou de tout le document.

**Fonction catalogue**

Cette fonction permet de connaître à tout instant les titres des documents, constantes ou imprimés mémorisés.

**Fonction mailing**

Le programme permet d'introduire jusqu'à 100 variables par fichier.

Le tri automatique s'opère sur 10 clés de sélection.

### CARACTÉRISTIQUES

- Micro-processeur Z 80 A - 4 MH
- PROM 16 K pour logiciel du texte
- RAM 32 K avec protection en cas de panne de courant extensible à 48 K
- ECRAN de visualisation 12 pouces : 20 lignes de 80 caractères. Caractères d'écriture de couleur verte
- Matrice de caractères 7 x 14 dans un plan de 9 x 15 cm
- Interface pour appareil à cassettes audio (480 signes/seconde)




Au Japon, il s'appelle MZ 1500, il fait déjà fureur, vous l'avez découvert de notre numéro 9. La version européenne s'appelle MZ 800 (seul un Japonais saurait vous dire pourquoi), elle adopte une couleur crème et un clavier où les caractères japonais ont été remplacés par les symboles graphiques bien connus des MZ 80A, K et 700.

Hormis la partie supérieure légèrement surélevée, la présentation extérieure du MZ 800 possède un indéniable air de famille avec celle de son grand frère, le MZ 700 : même clavier, même disposition des touches de fonction et de déplacement du curseur. Cette ressemblance confirme le souci de SHARP d'assurer une continuité matérielle et logicielle entre ses différents MZ ; continuité confirmée par le fait que le MZ 800 est compatible à 100% avec les logiciels conçus pour MZ 700.

Alors, quoi de nouveau ? Tout, serai-je tenté de vous répondre car, au-delà de cette compatibilité avec ses prédécesseurs, le MZ 800 est une machine entièrement nouvelle, totalement repensée par ses concepteurs. La MEMOIRE, d'abord : les 64 Ko de RAM disponibles en version de base supportent le BASIC et son moniteur et laissent 25 Ko aux programmes de l'utilisateur ; ils sont complétés par un second moniteur en ROM et par un I.P.L. (Initial Program Loader) permettant de charger et de lancer automatiquement un programme à partir d'un support quelconque (cassette, quick disc ou disquette). Une dernière mémoire ROM se charge de gérer toute erreur due à l'utilisateur, et cela, quel que soit le langage utilisé (moniteur, basic ou autre langage). A la mémoire vive initiale de 64 Ko peut être ajoutée une seconde mémoire de 64 autres Ko, appeléee RAM FILE. Elle remplit les mêmes fonctions qu'une cassette ou disquette, il est, en effet, possible de lire ou de stocker instantanément sur cette mémoire tout programme ou ensemble de données utilisables dans une application ; le transfert en mémoire centrale s'effectue en quelques millièmes de secondes. En version de base, 16 Ko de RAM sont ajoutés pour gérer la mémoire écran ; cette mémoire vidéo peut être doublée, en option, pour augmenter les capacités graphiques du MZ.

Si nous ajoutons encore 2 petits Ko de ROM pour générer les caractères, nous obtenons un



système capable de gérer simultanément jusqu'à 176 Ko de mémoire pour une même application... Record absolu pour un micro-ordinateur individuel.

Côté périphériques, le MZ 800 peut être connecté à 3 types de mémoires de masse différentes :

- Cassette standard avec système de transfert spécifique SHARP (1200 bauds) dont la fiabilité a largement fait ses preuves sur les différents MZ.
- QUICK DISC adapté aux disquettes 2, 8 pouces double face de 2*64 Ko de capacité. Fiable, économique, rapide, le délai de lecture ou d'enregistrement n'excède jamais 8 secondes.

Le QUICK DISC prend éventuellement la place du magnétophone sur le MZ.

- Simple ou double disquettes 5 pouces ayant chacune une capacité de 280 Ko formattés .

4 types différents d'imprimante peuvent fonctionner sous le contrôle du MZ 800 :

- MZ 1P 16 : Imprimante table traçante 4 couleurs sur papier de 110 mm bien connue des possesseurs de MZ 700.
- MZ 80 P5 : Imprimante à aiguilles 80 colonnes, 100 caractères/secondes capable d'imprimer tous les caractères graphiques propres à la série MZ.
- Toutes imprimantes au standard CENTRONICS, le MZ 800 d'origine possédant sa propre interface CENTRONICS.
- Toutes imprimantes au standard RS 232C grâce à une carte optionnelle intégrée au MZ.

En résumé, 95% des imprimanttes actuellement disponibles sur le marché sont connectables sur un MZ 800.

Terminons ce chapitre « HARD » en précisant que le MZ 800 possède 2 connecteurs capables de recevoir des JOYSTICKS compatibles ATARI (les plus courants) ; que la carte RS 232C lui permet de dialoguer avec tout système informatique muni de ce standard de transmission et qu'enfin, il peut émettre ses images vers un téléviseur couleurs via une prise PERITEL ou vers un miniteur vidéo N/B.

Mon crémier me le répète souvent : « Le HARD sans le SOFT, mon bon monsieur, c'est comme un repas sans fromage » ; heureusement, de ce côté, nous sommes également gâtés ; le BASIC IZ013 (vous l'auriez deviné) tient ses promesses et exploite à fond toutes les possibilités du MZ. Côté écran d'abord : l'affichage se fait sur 25 lignes de 40 ou 80 caractères

LES NOUVEAUTES
LES NOUVEAUTES

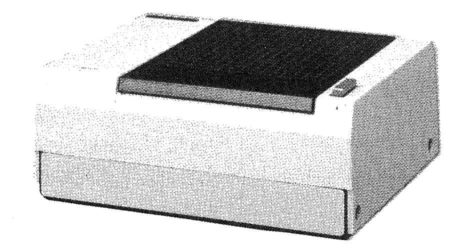
en 16 couleurs possibles ; un caractère peut être affiché dans n'importe quelle direction et peut avoir une taille quelconque ; toute la surface de l'écran si vous le désirez. Le mode haute résolution permet d'adresser séparément 640*200 points en 2 couleurs ou 320*200 points en 4 couleurs. La mémoire vidéo supplémentaire permet 640*200 points en 4 couleurs et 320*200 en 16 couleurs.

De nombreuses commandes basic permettent une infinie variété de créations graphiques qui ne sont limitées que par votre créativité et votre goût artistique. Ce goût artistique, vous pourrez l'exploiter également dans un tout autre domaine, grâce au générateur sonore intégré au MZ. Celui-ci est capable d'interpréter votre partition sur 7 octaves et 3 voies séparées et simultanées. Un générateur de « bruits blancs » viendra ajouter les effets les plus divers à vos créations sonores et vous permettra les imitations les plus folles : mitrailleuses... tirs intergalactiques... parfaite imitation des hurlements nocturnes de votre chère progéniture... etc. Côté efficacité, le Basic SHARP a toujours été un modèle de rapidité (boucle de 1 à 10 000 en 7 sec.), il s'est orné d'un ELSE dans ses boucles FOR-NEXT et de nombreuses commandes utiles au programmeur comme, par exemple SEARCH qui permet de retrouver une chaîne de caractères dans un programme basic. Son gros avantage réside toujours dans le fait qu'il est logé en mémoire vive, ce qui permet, d'une part, de le modifier à volonté, ou, d'autre part, de le remplacer purement et simplement par un autre langage sans empiéter sur la mémoire utilisateur.

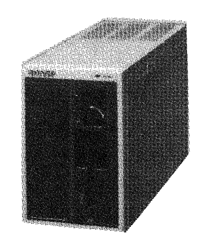
De nombreux autres langages seront d'ailleurs très bientôt disponibles dont un LOGO spécialement développé par SHARP pour le MZ 800. Dernier point très important pour conclure ce chapitre SOFT : le système d'exploitation CP/M est dès maintenant disponible sur cette machine et lui permet ainsi l'accès à une multitude de programmes professionnels (utilitaires, gestions de données, traitements de textes, etc.). Les concepteurs du MZ 800 ont voulu un système fiable, performant, universel, parfaitement adapté aux exigences sans cesse croissantes et diversifiées de ses utilisateurs : gageons que cette recette assurera le succès amplement mérité auquel aspire le « petit dernier » de la gamme MZ.

S.B.

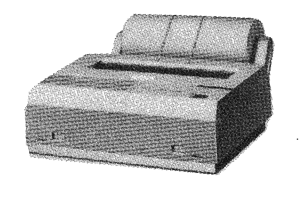


QUICK DISC.



DISQUETTE 5 POUCES



IMPRIMANTE - TABLE TRAÇANTE



# SHARP ENSEIGNEMENT




**SHARP BUREAUTIQUE**

**L'avance technologique au service de l'enseignement.**

**Pionnier des calculatrices scientifiques et des micro-ordinateurs de poche, Sharp propose aux professeurs de maths et de physique un programme enseignement particulièrement adapté aux nouvelles méthodes pédagogiques et à l'évolution permanente de la technique informatique.**
**Ce programme accorde tout d'abord un tarif préférentiel sur le prix de vente normalement pratiqué sur 14 références et options de la gamme Sharp spécialement choisies pour leur parfaite adéquation aux cours ainsi que pour les enseignants une inscription gratuite au club des utilisateurs de micro-ordinateurs Sharp : Le club des Sharpentiers.**

Le stand SHARP ENSEIGNEMENT au salon EDUCATEL




Ce communiqué et son « logo », vous le retrouverez souvent, si vous êtes professeur ou étudiant, dans vos revues spécialisées. Au delà d'une banale campagne publicitaire, il confirme le désir constant de la société Sharp de ne pas se cantonner dans un rôle passif de fabriquant et de distributeurs de produits, si parfaits soient-ils, mais d'entretenir une politique dynamique de communications entre elle même et les différentes catégories d'utilisateurs de produits Sharp. Ce bulletin n'en serait-il pas la preuve ?
Ce programme enseignement répond à un besoin croissant du corps enseignant de nouveaux outils fiables, performants, maniables, favorisant une ouverture d'esprit vers de nouvelle méthodes pédagogiques ; ainsi qu'à une volonté de rapprochement entre l'éducation nationale et le monde professionnel.
Il concerne tous les professeurs d'enseignement secondaire et supérieur de lycées, collèges, L.E.P. et I.U.T.

## EN QUOI CONSISTE CE PROGRAMME ?

- **Sharp** a mis en place, dans toute la France, un réseau de points de vente privilégiés disposant de tous les matériels, informations documentations ; le « logo » Sharp enseignement les rend facilement reconnaissables.
- **Sharp** enverra à tout professeur de maths et de physique un dossier très documenté comprenant une explication détaillée du fonctionnement du programme enseignement, la liste des points de vente enseignement périodiquement mise à jour, une documentation complète sur les produits micro-informatique et scientifiques Sharp et un carnet de chèques...
- **Les points de vente** enseignement seront habilités, en échange d'un chèque enseignement, à pratiquer un tarif très préférentiel sur l'achat de l'un des 14 produits concernés par le programme. Tout professeur ou élève, en possession d'un chèque enseignement peut bénéficier de ce tarif.
- **Le club des Sharpentiers** offrira à tout professeur qui aura acquis une machine au tarif enseignement, un abonnement gratuit de 1 an au bulletin du club (6 numéros).

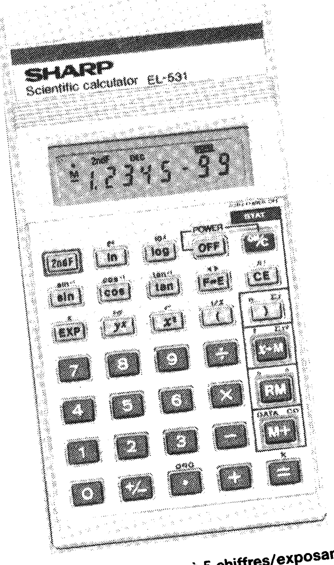
Le programme enseignement n'est pas limité dans le temps ; il sera ponctué, au fil des mois, de nouvelles actions favorisant le contact entre SHARP et le monde de l'enseignement : concours, rencontres, échanges. Professeurs, vous êtes nombreux au club, pourquoi ne pas nous soumettre vos idées ?

## Les produits Sharp enseignement

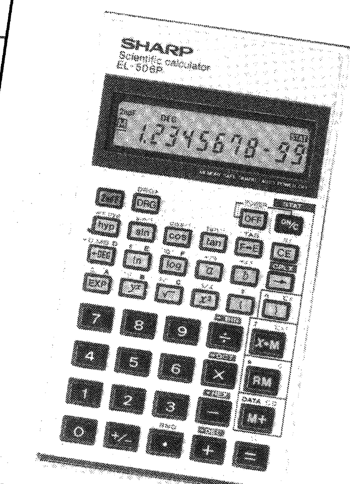
**MICRO-ORDINATEURS**
PC 1246
PC 1251
PC 1260
PC 1261
PC 1401
PC 1350
PC 1500A

**PERIPHERIQUES**
CE 150
CE 125
CE 126P

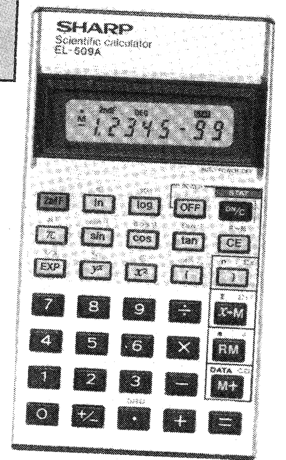
**CALCULATRICES SCIENTIFIQUES**
EL 531
EL 509
EL 506 P
EL 512




**EL-531 (8 chiffres [mantisse à 5 chiffres/exposant à 2 chiffres])** * 15 niveaux de parenthèses et 4 niveaux d'opérations en cours * Environ 10.000 heures d'usage avec 2 piles * 34 fonctions pré-programmées, y compris fonctions statistiques et interrogation de mantisse * Mémoire à 3 touches.
LCD FL M APO √x K 1/x % x²




**EL-506P (10 chiffres)** * Calculatrice format portefeuille avec 56 fonctions scientifiques * Calcul trigonométrique et hyperbolique et leurs inverses * Fonctions statistiques pré-programmées * Protection de la mémoire * Binaire, octale, décimale, hexadécimale.
LCD FE FL M APO √x K 1/x Δ% x²




**EL-509A (8 chiffres [mantisse à 5 chiffres/exposant à 2 chiffres])** * Entrée de formules * 3 niveaux de parenthèses et 3 niveaux d'opérations en cours * 32 fonctions pré-programmées * Touche statistiques.
LCD FL M APO √x K 1/x Δ% x²

**EL-512 (10 chiffres [mantisse à 8 chiffres, exposant à 2 chiffres])** * Notation directe et réserve multi-formule * 15 niveaux de parenthèses et 8 niveaux d'opérations en cours * 61 fonctions pré-programmées * Conversion décimal/hexadécimal * 9 mémoires * 128 pas de programme.
LCD FE FL M SC MS APO √x K 1/x x²

**SI VOUS ÊTES PROFESSEUR ET DÉSIREZ RECEVOIR NOTRE DOSSIER ENSEIGNEMENT, OU LE FAIRE RECEVOIR PAR L'UN DE VOS CONFRÈRES, REMPLISSEZ LE COUPON CI-DESSOUS.**

Nom : ........................................................................
Prénom : ........................................................................
Professeur de : ........................................................................
Etablissement : ........................................................................
Adresse : ........................................................................
Code postal : ........................................................................
Ville : ........................................................................
Désire recevoir rapidement le dossier enseignement **Sharp.**

DATE :
SIGNATURE :




**Les affaires club sont essentiellement constituées de matériels, accessoires ou périphériques qui ne sont pas (ou plus) commercialisés par le réseau de revendeurs SHARP. Comme vous pouvez le constater, leurs prix sont très attractifs. Une précision, toutefois, seules, les « Affaires club » sont commercialisées par le club, à l'exclusion de tous autres matériels normalement distribués par les points de ventes SHARP.**

MZ 80 B

| | | |
|---|---|---|
| MZ 80 MDB | Master disquette + manuel Disc-BASIC SB 6510 | 150 F |
| MZ 80 BD02 | Disquette F. DOS contenant tous les utilitaires nécessaires à la programmation en ASSEMBLEUR ; accompagnée d'un superbe classeur contenant plus de 300 pages d'explications. | 600 F |
| MZ 80 BT03 | Compilateur BASIC sur cassette avec notice à recopier sur la disquette F DOS. | 250 F |
| MZ 80 BD03 | Master disquette BASIC SB 6511 spécialisé dans les transmissions sous RS 232 ou CENTRONICS. | 150 F |
| MZ 80 GM | Carte mémoire graphique page 1. | 700 F |
| MZ 80 GMK | Carte mémoire graphique page 2 complémentaire à la carte GM. | 500 F |
| MZ 80 AB01 | Cassette de conversion des programmes BASIC MZ 80 K en MZ 80 B. | 70 F |

MZ 80 K

| | | |
|---|---|---|
| MZ 80 I/01 | Carte universelle d'interface entrées/sorties. | 300 F |
| MZ 80 T20C | Cassette et notice pour la programmation en langage machine. | 150 F |

| | | |
|---|---|---|
| MZ 80 TF | Manuel d'utilisation, en français du MZ 80 K, très pédagogique, il est universel pour l'ensemble des BASIC's MZ. | 100 F |

MZ 80 A

| | | |
|---|---|---|
| MZ 80 AT01 | Cassette de conversion des programmes BASIC MZ 80 K en programme MZ 80 A. | 70 F |

## BON DE COMMANDE

NOM .................................... PRENOM ..........................

ADRESSE ......................................................................

Je passe commande de :

| REF. | QUANT. | DESIGNATION | P.U TTC | P.TOTAL TTC |
|---|---|---|---|---|
| | | | | |
| | | | TOTAL TTC | |

Ci-joint mon règlement de ....... F., par chèque bancaire ou C.C.P. établi à l'ordre du CLUB DES SHARPENTIERS. Il représente le montant total TTC de ma commande. Je prends bonne note qu'au cas ou ma commande ne pourrait être honorée dans la limite des stocks disponibles, je serai intégralement remboursé du montant des articles non livrés.

DATE ........................................ SIGNATURE

# BULLETIN DES ANCIENS COMBATTANTS

## (ou les mémoires d'un fondateur)

**Aurais-je pu me douter en insérant, un beau matin de 1980, le premier bulletin d'inscription au club des Pécédouzeonsistes dans la pochette du premier ordinateur portable du monde le SHARP PC-1211, que 4 ans plus tard ce simple geste aurait permis d'aboutir à un résultat aussi magnifique que l'est le club des Sharpentiers.**

En effet, que de chemin parcouru depuis les 4 pauvres feuilles photocopiées de notre premier journal jusqu'à cette somptueuse revue dont vous vous délectez **tous les 2 mois**

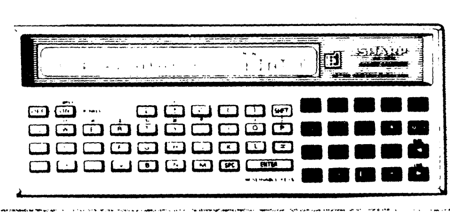
A ce propos je remercie Sylvain Bizoirre d'avoir pensé à rééditer les 5 premiers numéros sous la forme d'une brochure. Voilà une très bonne idée qui permettra certainement à quelques unes de plus belles pages de la littérature française de ces dernières années de passer à la postérité. Bravo ! Et puisque j'évoque le nom de notre cher rédacteur en chef, que de chemin là aussi parcouru entre le jeune passionné famélique qui rodait dans les couloirs de chez SHARP à la recherche d'un octet à ronger et le dynamique animateur qu'il est devenu, entouré d'une armée de secrétaires et de rédacteurs dans son somptueux bureau d'Aubervilliers, il est la véritable tour de contrôle * de ce club, symbole vivant de l'opulence et de la réussite. Alors me direz-vous, si la réalité est si plaisante pourquoi avoir abandonné ce poste aussi confortable ? Et bien parce qu'il faut savoir retourner à ses premières amours, (en ce qui me concerne, c'est le commerce), lorsque l'on sait que la relève est assurée. Mon passage au club m'aura au moins conforté dans une idée c'est que le véritable ordinateur de demain c'est le Pocket Computer (et là je suis sérieux).

C'est cette idée que je m'efforce de faire passer auprès de toutes les entreprises, et à ce propos, je lance un appel dans ces colonnes afin que tous les concepteurs de logiciels professionnels me contactent dans le but de créer une banque de programmes susceptibles d'aider à la promotion des Pockets dans le monde du travail.

D'avance merci.

A dans 4 ans.

Didier Monternot



**RECTIFICATION**

Dans le numéro 1 j'annonçais la sortie du bulletin numéro 2000 en mars 2231, je vous demande bien vouloir vous même rectifier les dates.

* La tour de contrôle signale un épais brouillard, au moment de la rédaction de cet article, entre le bureau de l'auteur et celui du club...

# ENQUETE SHARPENTIERS

Votre NOM : .......................... PRENOM : .......................... AGE ..........................
PROFESSION (en précisant votre domaine d'activités) : ..........................
DEPARTEMENT DE RESIDENCE : ..........................

MATERIEL(S) POSSEDE(S) : ..........................
PERIPHERIQUE(S) : ..........................
UTILISATION : JEUX : .......................... oui ☐ non ☐
INITIATION A L'INFORMATIQUE : .... oui ☐ non ☐
ENSEIGNEMENT : .................. oui ☐ non ☐
PROFESSIONNEL : ............... non ☐
oui ☐ (précisez le domaine d'utilisation) ..........................
..........................

AVEZ-VOUS DEJA FAIT APPEL AUX SERVICES DU CLUB (logiciels, renseignements, affaires club, courrier, téléphone) : oui ☐ non ☐
SI OUI, EN AVEZ-VOUS ETE SATISFAIT : oui ☐
oui mais ☐
non ☐ pourquoi : ..........................
AIMERIEZ-VOUS VOIR D'AUTRES SERVICES PROPOSES PAR LE CLUB : oui ☐ non ☐
SI OUI, LESQUELS : ..........................
..........................

ETES-VOUS DEJA PASSE NOUS VOIR LE MERCREDI APRES-MIDI : oui ☐ non ☐
SI OUI, AVEZ-VOUS ETE SATISFAIT DE CETTE (CES) VISITE(S) : oui ☐
non ☐ pourquoi : ..........................
..........................

L'INSCRIPTION ANNUELLE AU CLUB (160 F) VOUS PARAIT-ELLE : PAS CHERE ☐ NORMALE ☐ TROP CHERE ☐
L'ACTUELLE PRESENTATION DU BULLETIN VOUS SATISFAIT-ELLE : oui ☐
non ☐ pourquoi : ..........................
..........................

LE BULLETIN EST-IL LU PAR D'AUTRES PERSONNES QUE VOUS-MEMES :
non ☐
oui ☐ COMBIEN ☐ FAMILLE ? ☐ RELATIONS ? ☐
TENANT COMPTE DES DIFFERENTES CATEGORIES DE MACHINES TRAITEES, TROUVEZ-VOUS LA PLACE RESERVEE A LA VOTRE :
INSUFFISANTE ☐ NORMALE ☐ TRES IMPORTANTE ☐
AIMERIEZ-VOUS VOIR DES RUBRIQUES DISPARAITRE DU BULLETIN : non ☐
oui ☐ lesquelles : ..........................
..........................

LE CONTENU DE NOS ARTICLES VOUS PARAIT-IL :
PAS ASSEZ TECHNIQUE ☐ SUFFISAMMENT TECHNIQUE ☐ TROP TECHNIQUE ☐

VOUDRIEZ-VOUS EN VOIR APPARAITRE D'AUTRES : non ☐
oui ☐ lesquelles : ..........................
..........................

LISEZ-VOUS (T)OUJOURS, (S)OUVENT, (R)AREMENT, (J)AMAIS LES RUBRIQUES SUIVANTES :

| | T | S | R | J |
|---|---|---|---|---|
| Applications professionnelles : .......... | ☐ | ☐ | ☐ | ☐ |
| Nouveautés : .......... | ☐ | ☐ | ☐ | ☐ |
| Micro-informatique professionnelle SHARP : .......... | ☐ | ☐ | ☐ | ☐ |
| Les articles consacrés à d'autres machines que la vôtre : .......... | ☐ | ☐ | ☐ | ☐ |

DE CES 2 FORMULES, LAQUELLE PREFEREREZ-VOUS, A L'AVENIR :
☐ Achat du bulletin chez votre libraire ou abonnement pur et simple (120 F)
☐ Formule actuelle : Abonnement + services club (160 F)

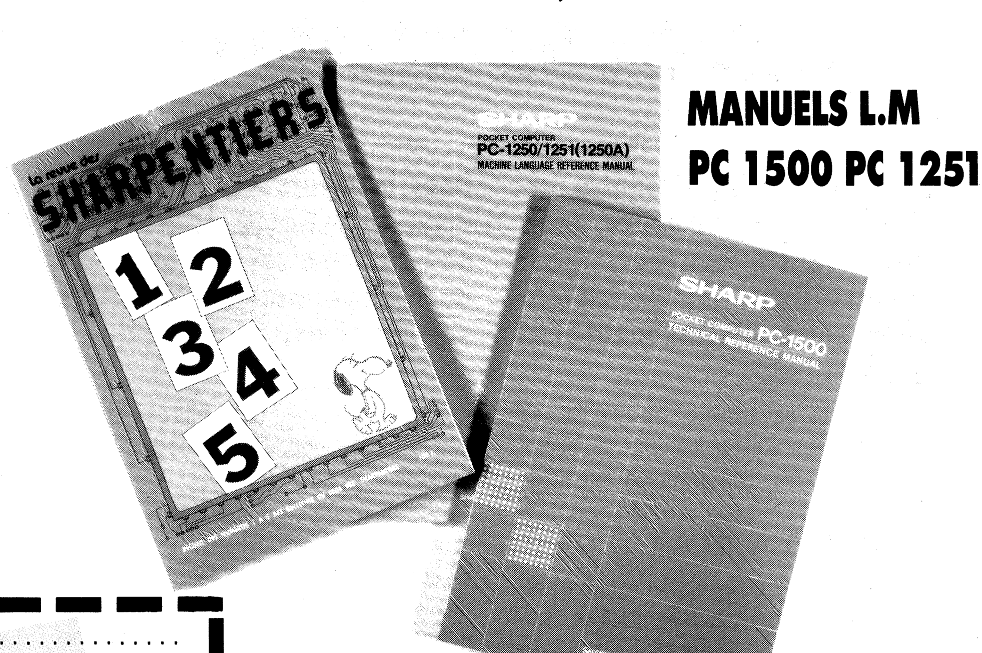
L'EXPERIENCE CLUB VOUS PARAIT-ELLE GLOBALEMENT POSITIVE : oui ☐ non ☐
VOS COMMENTAIRES : ..........................
..........................
..........................

Merci de bien vouloir retourner ce questionnaire au Club avant le 30 mars 1985.

# DANS VOTRE BIBLIOTHEQUE...

## LE SHARPENTIER

Entre PROUST et VOLTAIRE, vous pourrez maintenant glisser votre volume du SHARPENTIER. Nous avons rassemblé, en un seul volume de 120 pages, la reproduction fidèle des numéros 1 à 5 du bulletin du club. Ce recueil vous fera participer à la naissance de notre club (1981) et vous fournira tous les programmes, astuces et connaissances acquis durant les 2 premières années d'activité des Sharpentiers. Bien au-delà, l'achat de ce volume représente un investissement durable que les bibliophiles du monde entier s'arracheront dans les siècles à venir...




## MANUELS L.M PC 1500 PC 1251

M. ........................................

ADRESSE ........................................

........................................

vous prie de bien vouloir lui faire parvenir

- ☐ Recueil(s) des n° 1 à 5 des bulletins SHARPENTIERS. au prix de 110 F. (port compris)
- ☐ Bulletin(s) SHARPENTIER n° 6
- ☐ Bulletin(s) SHARPENTIER n° 7 AU PRIX DE 20 F. CHACUN
- ☐ Bulletin(s) SHARPENTIER n° 8
- ☐ Bulletin(s) SHARPENTIER n° 9
- ☐ Bulletin(s) SHARPENTIER n° 10
- ☐ Manuel(s) Langage Machine PC 1500
- ☐ Manuel(s) Langage Machine PC 1251 au prix de 167,40 F. l'un (port compris).

Je sais que ce livre est en anglais et qu'il s'agit d'un manuel de référence

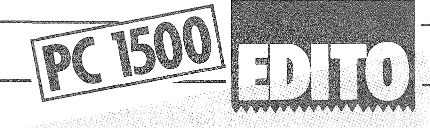
Ci-joint un chèque de ............ francs.

Date

Signature

Disponibles en exclusivité au club des Sharpentiers, vous pouvez vous procurer ces 2 manuels soit par correspondance au prix de 167,40 F. l'un (port compris) soit directement au club, le mercredi après-midi (exclusivement) au prix de 150 F.

Ces manuels, en anglais très bien documentés, sont des ouvrages de référence et n'abordent par l'initiation au langage machine.




**Des programmes courts, simples, précis. Voilà ce que nous aimons voir dans les colonnes du bulletin. Parfois, ce souci de précision vous amène a réaliser des exploits. Vous serez aussi étonné que nous par le programme de M. Pierson qui en trois lignes de basic réalise un jeu d'arcade fort connu : BOMBER ; ou, par le programme de Mr Nicolas qui permet des tris alpha très très rapides. Envoyez-nous, vous aussi, vos petits programmes qui reflètent votre maîtrise du pocket.**

**A bientôt.**

**Luc Bureller**

# GESTION DU CLAVIER (1)

**Le clavier est le partenaire privilégié du programmeur, mais aussi de l'utilisateur. C'est pourquoi il convient de bien connaître son fonctionnement et sa gestion.**

**Pour la scrutation clavier, on dispose, en BASIC, de INKEY$ ; en L.M., ce n'est pas tout à fait aussi simple mais les possibilités sont énormes.**

Commençons par explorer les routines les moins évoluées, c'est-à-dire celles se rapprochant de plus près de la structure interne de la machine. (Quelques rudiments de L.M. sont nécessaires pour comprendre cette explication.)

## STRUCTURE D'UN CLAVIER

Il faut déjà savoir qu'un clavier est un entrecroisement de conducteurs composé de 2 trames ; l'une verticale, l'autre horizontale ; chaque intersection correspond à une touche du clavier. Lorsque l'on appuie sur cette touche, le contact s'établit, et, si un courant passe dans l'un des conducteurs, on le retrouve dans le conducteur perpendiculaire à celui-ci.
Comptons les touches de notre clavier ; nous en trouvons 65 ; BREAK/ON n'étant pas traité comme les autres, il nous en reste 64, soit 64 entrecroisements de 2 fils, donc : 8 fils d'entrée et 8 fils de sortie. Le schéma classique en carré est une simplification car le constructeur préfère agencer les touches dans un ordre plus agréable pour le programmeur ; les conducteurs ne sont, alors, plus alignés, mais le principe reste le même.

La trame exacte du clavier PC1500 est reproduite en figure 1.

| | PAO | 1 | 2 | 3 | 4 | 5 | 6 | PA7 |
|---|---|---|---|---|---|---|---|---|
| IN0 | 2 | • | 1 | ) | + | = | » | 3 |
| IN1 | 5 | – | 4 | L | * | « | MODE | 6 |
| IN2 | 8 | OFF | 7 | O | / | P | Cl | 9 |
| IN3 | 4 | S | J | K | D | F | A | G |
| IN4 | SHIFT | F1 | F5 | F6 | F2 | F3 | DEF | F4 |
| IN5 | Y | W | U | I | E | R | Q | I |
| IN6 | N | X | M | ( | C | V | Z | B |
| IN7 | | | 0 | ENTER | RCL | SPACE | SML | |

Tiré du « Technical Reference Manuel ».

La routine en ROM située à l'adresse E418 va nous être d'un grand secours pour savoir quelles sont les touches pressées.
Le désassemblage de cette routine nous est fourni en figure 2 ; nous allons juste étudier les résultats qu'elle fournit et non son fonctionnement interne.
Un A est utilisé en entrée pour indiquer les colonnes testées. Il faut décomposer A, bit à bit pour savoir quelles vont être les colonnes sélectionnées. Le bit O de A enverra un courant dans la colonne PAO s'il est à 1 ; cette colonne sera donc testée. Il en va de même pour les 7 autres bits.
La routine E418 force A à &FF, donc, tous les bits sont allumés et tout le clavier est testé.
La routine E41A conserve A tel qu'il était lors de l'appel par SPJ. La sortie s'effectue selon le même format que l'entrée, le résultat se trouvant dans A. Les bits allumés correspondent à une touche appuyée à l'intersection de la ligne désignée par le bit allumé en sortie, et à la colonne d'un bit allumé en entrée. Si plusieurs colonnes sont désignées lors du traitement et qu'un bit est allumé en sortie, on ne pourra pas dire quelle est la touche appuyée, mais seulement qu'au moins une touche de cette rangée est appuyée.
Au retour de cette routine, Z = 1 si aucune touche n'est appuyée pour le A d'entrée, et A vaut Zéro. Pour tester tout le clavier, on fera SPJ E418. Si, au retour, Z = 1 : Aucune touche n'est appuyée.

**EXEMPLE 1 :**
E appuyé ? Colonne PA4 = 00010000 = & 10
Rangée IN5 = 00100000 = & 20
LDI A,&10
SPJ E41A
BII A,&20 : Z = 0 : Si bit 5 de A allumé : « E » appuyé.

**EXEMPLE 2 :**
ou appuyé ?
Colonne PA7 = 10000000 = &80
Rangée IN7 = 10000000 = &80
Colonne PAO = 00000001 = &01
Rangée IN7 = 10000000 = &80

Cela revient à tester si, pour les colonnes PAO ou PA7, la rangée IN7 est à 1.
LDI A,&81 (&80 pour et &01 pour)
SPJ E41A
BII A,&80 : Z = 0, si bit 7 de A allumé : ou appuyé.

Dans le même principe, on peut tester ON/BREAK en faisant SPJ E451, ou, plus simplement VMJ A6, puisque c'est une macro-instruction. Au retour, Z = 0 si BREAK est enfcncé.
Vous pouvez déjà désassembler la routine E42C qui est l'équivalent de INKEY$, vous avez maintenant les informations nécessaires ; nous le développerons néanmoins dans un prochain article où nous étudierons un SUPER INKEY$ qui renverra toutes les touches appuyées au clavier, nous verrons également les erreurs de INKEY$.

Marc GIRONDOT.

```
E418.LDI  A,FF
E41A:STA# (F00C)
E41E.ANI# (F00E),0
E423.LDI  UL,03
E425.LOP  E425
E427.ITA
E429.EAI  A,FF
E42B.RTN
```

```
E418.B5FF
E41A.FDAEF00C
E41E.FDE9F00E00
E423.6A03
E425.8802
E427.FDBA
E429.BDFF
E42B.9A
```

Désassemblage de la routine en E418-E42B (SOFT MONITOR)

# OPTIMISATION DES CE 159 ET CE 161

**Les deux modules CE-159 & CE-161, s'ils permettent tous les deux d'étendre la MEV utilisateur (respectivement de 8Ko et de 16 Ko, c'est-à-dire 10042 et 18234 octets pour le PC-1500, 14138 et 22330 octets pour le PC-1500A), ont surtout l'avantage de pouvoir conserver les programmes hors de l'ordinateur, lorsqu'ils sont utilisés en « zone d'extraction exclusive ».**

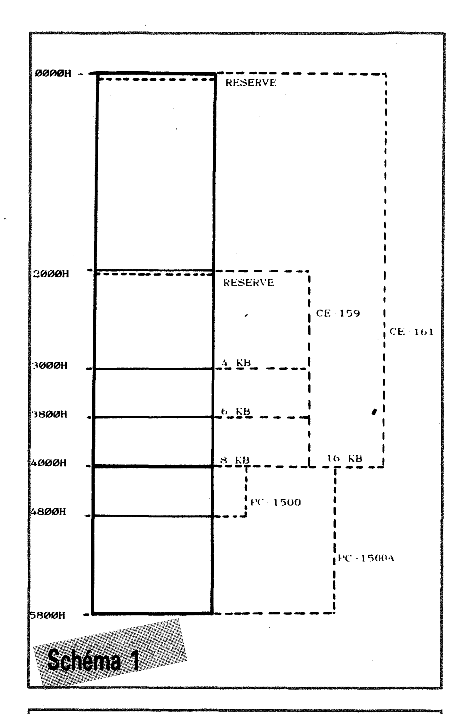


Schéma 1

Cette expression « barbare », tirée des manuels Sharp, désigne la procédure qui permet :

• L'écriture de programme et de contenu réservé pour stockage.

• La protection contre l'effacement ou la modification du programme stocké et du contenu réservé.

Cette zone pour utilisation exclusive en extraction, qui peut-être sélectionnée, pour le CE-159, à l'aide du commutateur de réglage de capacité :
4 Ko (3767 octets de programme)
6 Ko (5815 octets de programme)
8 Ko (7863 octets de programme)
ne peut-être pour le CE-161 que de :
16 Ko (16056 octets de programme).
Le commutateur de réglage est alors sur la position de mise en fonction.
(voir SCHEMA 1)
Lorsque le module doit être utilisé en zone d'extraction exclusive :

**QUE DOIT-ON FAIRE ?**
**QUE SE PASSE-T-IL ?**

**1.** Placer le commutateur sur la position de dégagement.
**2.** Indiquer la capactié, pour le CE-159, (4,6 ou 8 Ko), en ayant évalué la capacité du programme à utiliser en extraction.
**3.** Couper l'alimentation.
**4.** Connecter le module, rétablir l'alimentation et faire :
(MODE PRO) —> (ou :) (CL) NEWO (ENTER)
A ce niveau, examinons la zone des pointeurs :
7860 FF
7861 FF
7862 FF
7863 XO
Partie haute d'adresse de haut de MEV.
7864 48
Partie haute d'adresse de bas de MEV. (58 pour le PC-1500A).
7865 XO Partie haute d'adresse début du basic.
7866 C5 Partie basse d'adresse début du basic.
**5.** Entrer le contenu des touches personalisables.
**6.** Entrer le programme BASIC.
**7.** Entrer le programme basic pour une utilisation exclusive en extraction.
Les adresses 7867 et 7868 pointent alors sur les parties haute et basse du basic.
**8.** Exécuter le programme pour extractions exclusives.
Le programme remplit la zone vide en écrivant FF à chaque adresse et écrit END à la ligne 65279.
Le SCHEMA 2 montre la façon dont la zone est alors organisée.
**9.** POKER les 8 octets en début de module.
Pour chaque module, on trouve la même organisation :

X000 55
1 XO Partie haute d'adresse haut de MEM.
2 00 Partie haute d'adresse début du basic en, considérant &0000 = adresse haut de MEM.
3 C5 Partie basse d'adresse fin du basic en considérant &000 = adresse haut de MEM.
4 ** Code fonction de la taille de la MEM. 1KB : « 04H » 2KB : « 08H » 4KB : « 10H » 8KB : « 20H » 16KB : « 40H ».
5 00 Non défini.

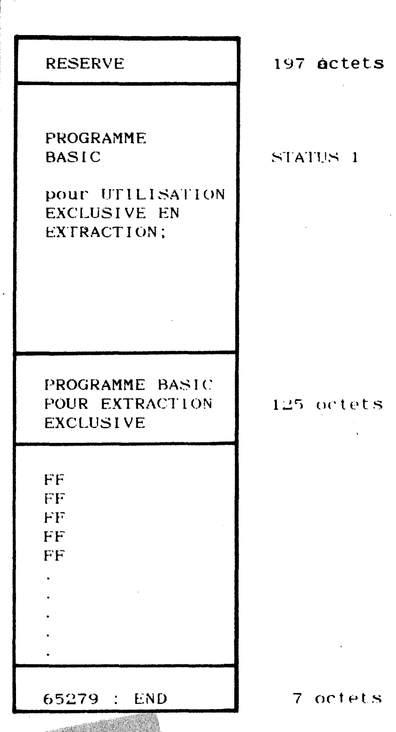


Schéma 2
Module de mémoire après la procédure d'extraction exclusive (commutateur en position de fonction)

6 00 Non défini.
7 FF ou « 00 » pour programme confidentiel.

**10** Couper l'alimentation et placer le commutateur de réglage sur la position de mise en fonction.

Lorsque l'alimentation est rétablie et après avoir entré :
(MODE PRO) —> (ou :) (CL) NEWO (ENTER) examinons à nouveau la zone des ponteurs :

7860 X0 Partie haute d'adresse haut de MEM.
7861 X0 Partie haute début basic en MEM.
7862 C5 Partie basse début basic en MEM.
7863 38 Partie haute d'adresse haut de MEV. (fonction du réglage de capacité pour CE-159)
7864 48 Partie haute d'adresse de bas de MEV. (58 pour PC-1500A).
7865 38 Partie haute d'adresse de début du basic *en MEV.*
7866 00 Partie basse d'adresse de début du basic *en MEV.*

Toutes ces manipulations sont longues et peu patiques.
• Le programme basic pour une utilisation exclusive en extraction reste dans le module.
• Une fois le commutateur de réglage sur la position de mise en fonction, le ou les programmes sont protégés définitivement de toutes modifications.
• La zone libre est perdue pour la programmation, mais surtout pour étendre éventuellement, la zone des variables.

| I | II | III |
|---|---|---|
| RESERVE | RESERVE | RESERVE |
| PC-LMPROMOD | PC-LMPROMOD | PC-LMPROMOD |
| PROGRAMME BASIC | PROGRAMME BASIC | PROGRAMME BASIC |
| ZONE LIBRE | | |
| ZONE POUR VARIABLES | VARIABLES | VARIABLES |
| Etat 1 Extraction exclusive | Etat 2 RUN ERROR 10 | Etat 3 Après CALL « PC LMPROMOD » déblocage |

**Schéma 3**
Utilisation « PC-LMPROMOD » pour déblocage de la zone libre

Le petit utilitaire propose, *en langage machine*, permet d'une part, de supprimer la plupart des manipulations précédentes et de les accélérer et d'autre part, de pouvoir, de façon immédiate, repasser de l'état MEM à l'état MEV.

Il devient ainsi possible, de modifier le programme protégé, de le sauver, éventuellement, sur bande magnétique et surtout de libérer la zone libre pour augmenter la taille de la zone des variables.
(voir SCHEMA 3) « PC-LMPROMOD » occupe 103 bytes. Il est relogeable, mais pour des raisons pratiques, il doit être implanté en début de module, en laissant libre la zone de réserve. &X0C6 (X = 0 pour CE-161 et 2 pour CE-159)

## 1 SAISIE DU PROGRAMME :

• Connecter le module.
• Mode RESERVE : (NEW).
• Mode PRO (CL) NEWO (ENTER).
• NEW &X12F (X = 0 pour CE-161 et 2 pour CE-159)
•Charger, si possible, un moniteur. Le « Moniteur HEXADECIMAL SHARP », donné dans le numéro 9 par exemple.
• Entrer les codes hexadécimaux de &X008 à &X0A3.
• Entrer « PC-LMPROMOD » de &X0C6 à &X12D.

## 2 PROCEDURE POUR EXTRACTION EXCLUSIVE :

• Pour CE-159, déterminer la capacité, en fonction de la taille du programme basic et modifier les adresses &X109 et &X115 en conséquence (voir programme).
• Entrer le programme basic.
• Mode RESERVE I : Appuyer sur OFF (F2).
•Couper l'alimentation du PC-1500.
• Mettre le commutateur en position de fonction.
• Rétablir l'alimentation.
• Faire (CL) NEWO (ENTER).

## 3 PROCEDURE POUR DEBLOCAGE DU MODULE :

• Mettre sous tension, le module étant connecté.
• Mode RESERVE I :Appuyer sur APO (F6).
• Mettre le commutateur en position de dégagement.
• Remettre sous tension.
• Mode RESERVE I : Appuyer sur ON (F4).

Le module est alors rétabli en état MEV, le programme est à nouveau accessible et la zone de variables peut occuper la zone libre dans le module.

Après modifications éventuelles ou utilisation, la procédure d'extraction exclusive peut-être à nouveau utilisée, comme décrite ci-dessus. Il est tout à fait possible d'implanter à la suite de notre utilitaire « PC-LMPROMOD », un programme en langage machine. Mais pour pouvoir être utilisé en extraction exclusive, celui-ci doit obligatoirement faire l'objet du (NEW &nnnn) d'usage.

Gérard MALLET

## ANNEXE 1

### LISTING DES CODES DES TOUCHES PERSONNALISABLES

(faire [NEW] en mode RESERVE)

| | | | | | | | | | CE-161 X=0 | CE-159 X=2 |
|---|---|---|---|---|---|---|---|---|---|---|
| X008 | 2A | 4F | 46 | 46 | 7E | 7B | FF | 7D | 82 | A2 |
| X010 | 2A | 2A | 4F | 4E | 7E | 7B | FF | 7D | 76 | 96 |
| X018 | 2A | 41 | 50 | 4F | 7E | 7B | FF | 7D | 97 | B7 |
| X020 | 2A | 2A | 2A | 2A | 2A | 2A | 2A | 2A | 70 | 90 |
| X028 | 2A | 2A | 50 | 41 | 47 | 45 | 20 | 4C | 05 | 25 |
| X030 | 49 | 42 | 52 | 45 | 2A | 2A | 2A | 2A | FA | 1A |
| X038 | 2A | 2A | 2A | 2A | 2A | 2A | 2A | 2A | 88 | A8 |
| X040 | 2A | 2A | 2A | 2A | 50 | 41 | 47 | 45 | 05 | 25 |
| X048 | 20 | 4C | 49 | 42 | 52 | 45 | 2A | 2A | 2A | 4A |
| X050 | 2A | 2A | 2A | 2A | 2A | 2A | 2A | 02 | 78 | 98 |
| X058 | F1 | 8A | 28 | F1 | 6F | 26 | 37 | 38 | F0 | 10 |
| X060 | 36 | 33 | 2A | 32 | 35 | 36 | 2B | 32 | ED | 0D |
| X068 | 31 | 31 | 29 | 40 | 04 | F1 | 8A | 28 | DA | FA |
| X070 | F1 | 6F | 26 | 37 | 38 | 36 | 33 | 2A | F8 | 18 |
| X078 | 32 | 35 | 36 | 2B | 32 | 38 | 38 | 29 | 0B | 2B |
| X080 | 40 | 06 | F1 | 8A | 26 | 45 | 33 | 33 | 12 | 32 |
| X088 | 46 | 40 | 00 | 00 | 00 | 00 | 00 | 00 | 0E | 2E |

## ANNEXE 2

### « PC-LMPROMOD »

(faire [NEW &X12F) en mode PRO avec X = 0 pour CE-161 et X = 2 pour CE-159)

X0C6 : X1 2F X1 2F X1 2F
Pointeurs sauvegardés

X0CC : FE FF 03 F1 8E 0D FF
Codes END

1. Routine pour extraction exclusive : (CALL &X0D3)

Sauvegarde des pointeurs module :

```
X0D3 : A5 78 63  LDA 7863H
   6 : 28        STA UH
   7 : 18        STA YH
   8 : 5A 00     LDI YL, 00H
   A : 6A 03     LDI UL, 03H
   C : B5 55     LDI A, 55H
   E : 51        SIN Y
   F : A4        LDA UH
  E0 : 51        SIN Y
   1 : FB        SEC
   2 : A5 78 65  LDA 7865H
   5 : A0        SBC UH
   6 : 51        SIN Y
   7 : A5 78 66  LDA 7866H
   A : 51        SIN Y
   B : B5 00     LDI A, 00H
   D : 88 05     LOP 05H
```

Sauvegarde des pointeurs Basic :

```
   F : 48 78     LDI XH, 78H
  F1 : 4A 65     LDI XL, 65H
   3 : A4        LDA UH
   4 : 18        STA YH
   5 : 5A C6     LDI YL, C6H
   7 : 6A 05     LDI UL, 05H
   9 : F5        TIN
   A : 88 03     LOP 03H
```

FF de STATUS 2 à 3F*F9H :

```
   C : A5 78 67  LDA 7867H
   F : 08        STA XH
X100 : A5 78 68  LDA 7868H
   3 : 0A        STA XL
   4 : 40        INC XL
   5 : B5 FF     LDI A, FFH
   7 : 41        SIN X
   8 : 4C 3F*    CPI XH, 3FH*
   A : 99 05     BZR, -05H
   C : 4E F9     CPI XL, F9H
   E : 99 09     BZR, -09H
```

Création ligne (65279 : END) :

```
  10 : A4        LDA UH
   1 : 08        STA XH
   2 : 4A CC     LDI XL, CCH
   4 : 58 3F*    LDI YH, 3FH*
   6 : 5A F9     LDI YL, F9H
   8 : 6A 06     LDI UL, 06H
   A : F5        TIN
   B : 88 03     LOP 03H
   D : 9A        RTN
```

2. Routine pour déblocage du module : (CALL &X11E)

```
211E : A5 78 63  LDA 7863H
  21 : 08        STA XH
   2 : 4A C6     LDI XL, C6H
   4 : 58 78     LDI YH, 78H
   6 : 5A 65     LDI YL, 65H
   8 : 6A 05     LDI UL, 05H
   A : F5        TIN
   B : 88 03     LOP 03H
   D : 9A        RTN
```

(*) : Valeur fonction de la taille de la MEM 4KB : « 2FH », 6KB : « 37H », 8KB et 16 KB : « 3FH ».

• Pour un programme non confidentiel : POKE &X007,&FF.

• Pour un programme confidentiel : POKE &X007,&00

# MONITEUR HEXADECIMAL (SUITE)

**Ce programme, publié dans notre bulletin N° 9, vous a, semble-t-il, causé beaucoup de soucis, et vous a valu un abondant courrier. La (savoureuse) lettre de Marc Rivet résume vos griefs et apporte les solutions nécessaires.**

**Une lettre... et on devient sharpentier-journaliste... pourquoi pas vous ?**

Besançon, le 17-10-84

Salut les Sharpentiers !,

Je viens de recevoir le N°9 de la revue du Club. Plusieurs articles consacrés à ma machine -le PC 1500- ont retenu mon attention, mais, étant grand amateur de L.M., je me suis précipité sur le "Moniteur Hexa-décimal" publié page 17.

Bon !, je rentre le programme pour obtenir..."erreur d'introduction"; pas de panique, ça ne tourne jamais du premier coup ! Je relis scupuleusement mes lignes de DATA... pas de problème. Rebelote... Même résultat ! Restons calme, il suffit de décortiquer le test de la ligne 4 : tout est O.K. sauf que chez moi, STATUS 1 vaut 1648. L'explication ? J'ai eu la mauvaise idée de mettre un "r" au mot "introduction", lettre qui avait été oubliée dans votre listing !

Après avoir beaucoup ri..., je rectifie le tir, et c'est reparti; j'obtiens enfin le "O.K." tant attendu. Travaillant sur PC 1500A + CE161, je lance donc CALL &C5, ce qui me donne un superbe "ERROR 0" (!).

Il est temps d'utiliser les grands moyens; je saute sur mon désassembleur-maison, MERGE la chose et contemple le résultat. Pas besoin d'aller plus loin : 1ère. instruction : "SJP 41AF" ! Que peut bien donner un appel de sous-programme échouant dans une zone de RAM réputée vide !!

L'explication est simple, l'étude du reste du désassemblage le démontre : contrairement aux assertions accompagnant votre listing, ce programme N'EST PAS RELOGEABLE dans la forme sous laquelle il a été publié (gênant pour un "programme d'usine" !). La solution la plus simple consiste, quelle que soit la configuration utilisée, à faire NEW &431F et laisser A=&40C5, dès lors, plus de problème, mais il est assez frustrant de sacrifier 16 Ko. dans ce but.

Il ne reste plus qu'à laisser le 1500 se charger du travail, voici comment :
- Ne pas modifier le commentaire joint au listing
- Supprimer le test de STATUS 1, ligne 4
- Modifier la ligne 5 comme indiqué ci-dessous
- Ajouter les lignes 280 et 290 également ci-dessous

Faire `RUN´ et la machine se chargera d'adapter le programme à la configuration utilisée !

Bien amicalement

Marc RIVET

```
5:X=PEEK &7863:W=X*256:FOR Y=0 TO 17:READ V:POKE W+V,PEEK(W+V)-&40+X:
  NEXT Y:PRINT "O.K.":END
280:DATA &C6,&E4,&E7,&EA,&FF,&121,&148,&161,&16C,&19D
290:DATA &1AB,&1CB,&1F6,&205,&244,&254,&2D8,&2EE
```

# LANGAGE MACHINE (5)

**Pour programmer en langage machine, il faut tout d'abord maîtriser les possibilités du microprocesseur, la manipulation des registres, mais surtout il faut bien connaître les routines disponibles en ROM. C'est ce que nous allons voir dans la suite de cette série sur le langage machine appliqué au PC 1500.**

Le PC 1500 dispose à lui seul de 16K de ROM et l'imprimante de 8K, on peut dénombrer environ 1 000 routines utiles !. Vu ce nombre il serait difficile de les expliquer toutes mais je vais tenter de vous décrire les plus importantes et les plus utiles. Pour cela nous classerons les routines selon quatre types :

**1)** les routines d'affichage
**2)** les routines de calcul
**3)** les routines de l'imprimante
**4)** les autres

Nous allons regarder cette fois çi les routines d'affichage. Ces routines seront très utiles à tous ceux qui désirent gérer l'écran à partir de programmes en langage machine.

## DU POINT DE VUE DE LA RAM

Il exsite en RAM système deux octets qui sont constamment utilisés par les routines que je vais vous décrire :

– (7875) = GCU position du curseur graphique (0-155)
– (787B) = CUR position du curseur texte (0-25)

Les routines utilisent aussi le BUFFER PRINT, tampon de 80 octets, qui va de 7B60 à 7BAF (en hexa).

Quand à la RAM écran, nous n'aurons pas à travailler directement dessus, ce qui n'est pas un tort. Les routines que nous allons voir s'occupent elles-mêmes de trouver les bonnes adresses dans cette zone qu'il n'est vraiment pas facile de gérer. A ce propos il faut aussi savoir que lors de l'utilisation des routines graphiques, le registre X (BC) du microprocesseur contient une adresse en RAM écran.

De même le registre U (HL) contient l'adresse d'une chaîne de caractères lors de l'affichage d'un texte.

Dans la suite de cet article, les routines sont données avec un pseudo mnémonique, l'adresse de la routine et si c'est le cas, entre parenthèses le numéro de la macro institution correspondante.

## ROUTINES D'USAGE GENERAL

Ces routines agissent sur les différents pointeurs du système.

| | | |
|---|---|---|
| HOME | ECAE | GCU-0 |
| CLS | EE71 (F2) | efface l'écran |
| STO | EDC1 | sauve l'écran en RAM système |
| RCL | EDD8 | rappelle ce qui avait été sauvegardé |
| PSHLCD | EE80 | sauve les caractéristiques de l'écran |
| POPLCD | EE99 | rappelle les caractéristiques de l'écran |

## AFFICHAGE EN MODE TEXTE

On a 3 instructions pour l'affichage en mode texte :

PRA pour afficher un seul caractère.
PRT pour afficher une chaîne de caractères.
PRX pour afficher le registre arithmétique.

| | | |
|---|---|---|
| PRA (X) | ED5B (8A) | Affiche le caractère de code ASCII A à la position X sur l'écran. |
| PRA (GCU) | ED57 | Affiche le caractère de code ASCII A à la position (GCU). |
| PRA (IGCU) | ED4D | Idem avec incrémentation de (GCU). |
| PRT (BUF) | ECFA | Affichage du buffer PRINT. |
| PRT | ED3B | Affiche la chaîne d'adresse U et de longueur A en début d'écran. |
| PRT (CUR) | ED00 (92) | Affiche la chaîne d'adresse U et de longueur A à la position (CUR). |
| PRX | EEEF | Affiche le contenu du registre arithmétique X à la position (CUR). |

## AFFICHAGE EN MODE GRAPHIQUE

Ici deux types d'instructions importantes :

POS traduit la position d'une colonne en une adresse en RAM
GPR affecte une valeur (de 00 à 7 F) à une colonne.

| | | |
|---|---|---|
| POS (GCU) | EE1F (8C) | X-adresse en RAM écran correspondant à la colonne (GCU) |
| POS A | EE22 | X-adresse en RAM écran correspondant à la colonne A. |
| GPR (GCU), A | EDEF | Affiche la colonne A en position (GCU), GCURSOR (GCU) : GPRINT A |
| GPR (IX),A | EDF6 (88) | Affiche la colonne A à la position X et incrémente X. |
| GPR (IX),(IY) | EDF5 | Affiche la colonne (Y) à la position X et incrémente X,Y. |
| INC (GCU) | EDB1 (8E) | incrémente (GCU) avec TLR. |
| TLR | EDAB (90) | Test la fin de l'écran, carry = 1 si le curseur graphique >155. |
| POINT | EECE | Au retour A contient la valeur de la colonne pointée par A. |

Nous allons à présent regarder ensemble la réalisation de quelques routines graphiques. Chacunes de ces routines vous sont données sous forme de mnémoniques, avec explications ligne à ligne, suivies des codes machines qui correspondent.

## INVERSION VIDEO :

Une routine simple mais instructive : Sur une boucle, on lit successivement les valeurs des

colonnes dont on prend le complémentaire. De plus cette routine pourra comme nous allons le voir servir à d'autres fins.
La routine s'écrit :

| | |
|---|---|
| LDi YH, 00 | on commence à la première colonne |
| LOOP LDA YH | A = position de la colonne |
| SJP POINT | A = POINT A |
| EOR A, 7F | on prend le complément à 7F (inversion) |
| STA YL | sauvegarde de la valeur dans YL |
| LDA YH | A = position de la colonne |
| SJP POS | X adresse correspondant à cette position |
| LDA YL | on récupère A |
| VMJ GPR (IX),A | et on fait un GPRINT |
| INC YH | on passe à la colonne suivante |
| CPi YH, &9C | on regarde si c'est la dernière |
| BCR, LOOP | si ce n'est pas la dernière, on boucle |
| RET | retour au basic |

Les codes machine sont :
58 00 94 BE EE CE BD 7F 1A 94 BE EE
22 14 CD 88 FD 50 5C 9C 91 14 9A 9A

## SCROLLING VERTICAL :

Pour changer un peu des SCROLL habituels (à faire tourner en boucle). On peut reprendre en grande partie la routine précédente, seul l'instruction EOR A, 7F est à changer :

| | |
|---|---|
| LDi YH,00 | on commence à la première colonne |
| LOOP LDA YH | A = position de la colonne |
| SJP POINT | A = POINT A |
| STA YL | sauve A dans YL |
| RLA | SCROLL avec CARRY (décalage à gauche) |
| LDA YL | récupère A |
| RLA | cette fois-ci, on a une rotation complète |
| STA YL | sauvegarde de la valeur dans YL |
| LDA YH | A = position de la colonne |
| SJP POS. | X adresse correspondant à cette position |
| LDA YL | on récupère A |
| VMJ IGPR (IX),A | et on fait un GPRINT |
| INC YH | on passe à la colonne suivante |
| CP YH, &9C | on regarde si c'est la dernière |
| BCR, LOOP | si ce n'est pas la dernière, on boucle |
| RET | retour au basic |

Les codes machine sont :
58 00 94 BE EE CE 1A DB 14 DB 1A 94 BE
EE 22 14 CD 88 FD 50 5C 9C 91 14 9A 9A

## INVERSION DE SENS :

Une routine amusante qui permet de lire à l'envers, l'écran de votre PC
Il faut pour cela procéder à une double inversion au niveau des pixels, de gauche à droite et de bas en haut.

| | |
|---|---|
| ANi (GCU), 00 | remise à zéro du curseur |
| LOOP LDA (GCU) | A = position du curseur |
| SJP POINT | A = POINT A |
| SJP INV | SPG d'inversion de A bit à bit |
| STA YH | sauve A dans YH |
| LDi A, 9C | A = 156 dec |
| SBC A, (GCU) | A = position symétrique |
| STA YL | sauve cette position dans YL |
| SJP POINT | A = POINT A |
| PSH A | empile cette valeur |
| LDA YL | récupère la position symétrique |
| SJP POS | X-adresse correspondante à la position |
| LDA YH | récupère YH |
| SJP GPR (IX), A | et premier GPRINT |
| POP A | dépile A |
| SJP INV | traitement de l'accumulateur |
| SJP GPR (GCU),A | deuxième GPRINT |
| LDA (GCU) | A = position du curseur |
| INC A | incrémentation |
| STA (GCU) | nouvelle valeur du curseur |
| CP A, 4E | regarde si milieu de l'écran |
| BCR LOPP | on continue si pas encore |
| RET | retour au basic |
| INV LDi UH, 08 | UH = 00 |
| LDi UL, 00 | UL = 08, on va boucler huit fois |
| LP 2 RLA | décalage à gauche |
| PSH A | empile A |
| LDA UH | A = UH |
| RRA | décalage à droite |
| STA UH | UH = A |
| POP A | récupère A |
| DJC LP 2 | boucle |
| LDA UH | A est devenu image dans un miroir de lui même, retour de la sous-routine. |
| RET | |

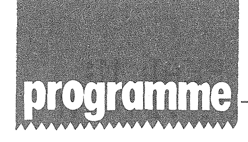
Les codes machine sont :
E9 78 75 00 A5 78 75 BE EE CE BE 25
34 18 B5 9C A1 78 75 1A BE EE CE FD
C8 14 BE EE 22 94 CD 88 FD 8A BE 25
34 BE ED EF A5 78 75 DD AE 78 75 B7
4E 91 2F 9A 68 00 6A 08 DB FD C8 A4
D1 28 FD 8A 88 0A A4 9A

Voilà donc en ce qui concerne l'utilisation des routines graphiques. Pour celles qui permettent d'afficher du texte, on se rapportera au programme LM du MICRO MONITEUR (bulletin n° 7), qui utilise la plupart de ces routines.
Vous trouverez ci-dessous un programme BASIC qui permet d'utiliser les trois routines graphiques, qui sont POKEs dans la zone des variables.
L'utilisation est simple, on entre un message et on choisi par KEY le type de la transformation.

Voilà, la prochaine fois nous regarderons de près les routines de calculs, et dès maintenant proposez-nous vos applications graphiques.

Pascal Abrivard

```
  1 "*************************************
  2 "*                                   *
  3 "*  EXEMPLE DE ROUTINES GRAPHIQUES LM *
  4 "*                                   *
  5 "*      (c) P. ABRIVARD et le CLUB   *
  6 "*                                   *
  7 "*************************************
  8 "
  9 "
 10 DATA &58,&00,&94,&BE,&EE,&CE,&BD,&7F
 15 DATA &1A,&94,&BE,&EE,&22,&14,&CD,&88
 20 DATA &FD,&50,&5C,&9C,&91,&14,&9A,&58
 25 DATA &00,&94,&BE,&EE,&CE,&1A,&DB,&14
 30 DATA &DB,&1A,&94,&BE,&EE,&22,&14,&CD
 35 DATA &88,&FD,&50,&5C,&9C,&91,&14,&9A
 40 DATA &E9,&78,&75,&00,&A5,&78,&75,&BE
 45 DATA &EE,&CE,&BE,&70,&B4,&18,&B5,&9C
 50 DATA &A1,&78,&75,&1A,&BE,&EE,&CE,&FD
 55 DATA &C8,&14,&BE,&EE,&22,&94,&CD,&88
 60 DATA &FD,&8A,&BE,&70,&B4,&BE,&ED,&EF
 65 DATA &A5,&78,&75,&DD,&AE,&78,&75,&B7
 70 DATA &4E,&91,&2F,&9A,&68,&00,&6A,&08
 75 DATA &DB,&FD,&C8,&A4,&D1,&28,&FD,&8A
 80 DATA &88,&0A,&A4,&9A
 85 "
 86 "
 90 CLEAR :DIM A$(0)*26:WAIT 0
100 FOR I=0TO 115:READ V:POKE &7050+I,V:NEXT I
110 A$(0)="-LE CLUB DES SHARPENTIERS-"
120 "M"CLS :INPUT "Message :";A$(0)
130 PRINT "1:VIDEO 2:SCROLL 3:ENVERS"
140 V=VAL INKEY$ :IF V<1OR V>3THEN 140
150 PRINT A$(0):ON VGOSUB 200,210,220
160 IF INKEY$ =""THEN 160
170 GOTO "M"
180 "
181 "
200 CALL &7050:RETURN
210 FOR I=0TO 23:CALL &7067:NEXT I:RETURN
220 CALL &7080:RETURN
```



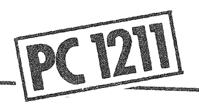



# DECATHLON

Tapez le programme en ne faisant, si possible, aucune erreur. Vous serez, alors, projeté aux Jeux olympiques en faisant : RUN - ENTER
Avant chaque épreuve est affiché le mode d'emploi, pour passer d'une ligne à l'autre, tapez ENTER.
Si l'imprimante est connectée, un tableau récapitulera les scores des 10 épreuves.
Le programme fonctionne sur un PC 1500 muni d'au moins 8 Ko.

Maurice CHOUCROUN

```
10 BEEP OFF :GOSUB 61E3:BEEP 2,75,200:D=4:RANDOM :DIM A$(7),B$(1)*24:A$="000068241F6444"
15 WAIT 0:CLS :CURSOR 3:PAUSE "*** 100 METRES ***":CLS
16 PAUSE "(4) ET (6) Pour avancer...":WAIT :PRINT :WAIT 0:CLS
20 GCURSOR 124:GPRINT 127:GCURSOR 0:GPRINT A$:GCURSOR 128:PRINT "PRET!"
22 GCURSOR 128:PRINT "     "
25 FOR I=100TO 0STEP -1:GCURSOR 128:PRINT I:NEXT I:BEEP 2,9:TIME =0:GCURSOR 128
30 IF INKEY$ ="4"LET H=1
40 IF INKEY$ ="6"AND HLET H=0:K=K+2:GCURSOR K:GPRINT A$:IF K>118GOTO 60
50 GOTO 30
60 S=40-TIME *2E4:BEEP 1:CLS :CURSOR 5:PAUSE "TeMPs ";TIME *1E4;" secondes":PAUSE
70 GOSUB 6E4:PAUSE :GCURSOR 8:BEEP 2:PAUSE "*** SAUT EN HAUTEUR ***":CLS
71 WAIT :PRINT "(ENTER) Pour s elancer":WAIT 0:CLS
72 G=1:WAIT 1:B$="7800606060606060"
75 FOR J=1TO 3:CLS :CURSOR 19:PAUSE "PRET!":PAUSE :CURSOR 19:PRINT "GO!  ":BEEP 1
80 GCURSOR 0:GPRINT A$:GCURSOR 100:GPRINT B$
90 FOR I=1TO 95STEP 2:GCURSOR I:GPRINT A$;:IF INKEY$ =CHR$ 13BEEP 1:GOTO 100
92 NEXT I
95 BEEP 3,80:CURSOR 19:PAUSE "SPLAT "+STR$ J:PAUSE :NEXT J:GOTO 200
100 I=I+6:ON GGOTO 110,120,130
110 IF I<95OR I>99GOTO 95
112 GOSUB 160:B$="7C00606060606060":GOTO 75
120 IF I<>97AND I<>99GOTO 95
122 GOSUB 160:B$="7E00606060606060":GOTO 75
130 IF I<>99GOTO 95
132 GOSUB 160:GOTO 200
160 GCURSOR 90:GPRINT "000000000000"
165 WAIT 50:GCURSOR 96:GPRINT "080E320A7E0767676A6868":GCURSOR 96:GPRINT "000000007812746878 7A7C68":WAIT 1
170 BEEP 3,5:CURSOR 0:PAUSE "BRAVO!!":PAUSE :G=G+1:RETURN
200 IF G=2LET S=S+7
201 IF G=3LET S=S+14
202 IF G=4LET S=S+20
205 GOSUB 6E4:A$(0)="00000000067E":A$(1)="0000060E3040":A$(2)="00060E102040"
206 A$(3)="060608102040":A$(4)="0C0C10202040":A$(5)="303020404040"
207 WAIT 0:BEEP 2:A$(6)="606040404040":CLS :CURSOR 1:PAUSE "*** LANCER DU POIDS ***":PAUSE
208 WAIT 0:PRINT "(4) Pour";:GPRINT I$;:PRINT " et (8) Pour";:WAIT :GPRINT F$:CLS :WAIT 0
210 CLS :GCURSOR 108:GPRINT "442A1B112E40"
212 G=0:FOR I=0TO 95STEP 5:GCURSOR I:GPRINT 127:NEXT I
215 TIME =0:BEEP 1:GCURSOR 100:GPRINT A$(0)
220 IF INKEY$ ="4"LET G=G+1:IF G>6LET G=6
230 IF INKEY$ ="8"LET G=G-1:IF G<0LET G=0
240 GCURSOR 100:GPRINT A$(G):IF TIME <>.0002GOTO 220
255 BEEP 1:G=2^G:IF G=1LET H=80
256 IF G=2LET H=40
257 IF G=4LET H=10
258 IF G=8LET H=20
259 IF G=16LET H=30
260 IF G=32LET H=40
261 IF G=64LET H=80
262 K=H-RND 10:FOR I=98TO KSTEP -1:GCURSOR I:IF POINT I=127LET J=I
263 GPRINT GOR POINT I;0+127*(POINT (I+1)>6)
265 NEXT I:CURSOR 20:BEEP 1:PAUSE STR$ ((100-K)/5)+" m":PAUSE
266 IF (100-K)/5>XLET X=(100-K)/5
267 E=E+1:IF E<>3GOTO 210
268 S=S+X:GOSUB 6E4:CLS :BEEP 2:PAUSE " *** SAUT EN LONGUEUR ***":PAUSE :CLS :X=0
269 WAIT :PRINT "(SPACE) Pour Prendre aPPui":PRINT "(ENTER) Pour sauter...":CLS
270 A$(0)="3012147F4404":A$(1)="1808691A040B0808":A$(2)="1868291A0C040B0808"
273 A$(3)="026E2838091E2828":A$(4)="6028386064782020"
279 FOR J=1TO 3:WAIT 0:CURSOR 5:PAUSE "PRET!":PAUSE :BEEP 1:CLS :FOR I=50TO 150STEP 5:GCURSOR I:GPRINT 64:NEXT I:GCURSOR
51:GPRINT 64
280 FOR I=0TO 43:GCURSOR I:GPRINT A$:IF INKEY$ =" "GOTO 300
285 NEXT I
290 GCURSOR 0:BEEP 1,20:PAUSE "SPLAT!":NEXT J:GOTO 350
300 WAIT 5:FOR R=0TO 4:GCURSOR I-1:GPRINT A$(R):IF INKEY$ <>CHR$ 13NEXT R:GOTO 290
302 WAIT 0:IF R<=2LET E=20*R+20:WAIT 4-R*2
304 IF R>2LET E=20*(4-R)+20:WAIT R
309 BEEP 1:GCURSOR I-3:GPRINT "602414684C1606":GCURSOR I:C$="000F0A0A1810":GPRINT C$
310 FOR L=ITO I+E:GCURSOR L:GPRINT C$
320 NEXT L:L=L+RND 5:BEEP 1,10:WAIT 10:GCURSOR L:GPRINT "3E2828204000":BEEP 1,10:GCURSOR L:GPRINT "785050406000"
340 GCURSOR L-6:PRINT " ":BEEP 1,30:CURSOR 0:PAUSE L/15;"m ":PAUSE :PAUSE
345 IF X<L/7LET X=L/7
346 NEXT J
350 S=S+X:GOSUB 6E4:H=0:G=0:K=0:L=0
360 CLS :BEEP 2:PAUSE "    *** 400 METRES ***":PAUSE :WAIT 0
362 WAIT :PRINT "(0),(.),(=) Pour avancer":WAIT 0:CLS
365 CLS :GCURSOR 0:GPRINT A$
370 GCURSOR 35:GPRINT 64:GCURSOR 60:GPRINT 64:GCURSOR 85:GPRINT 64
380 GCURSOR 110:GPRINT 127:CURSOR 19:PAUSE "PRET!":PAUSE :BEEP 1:TIME =0:CURSOR 19:PRINT "GO!"
390 IF INKEY$ ="0"LET H=1
392 IF INKEY$ ="."AND HLET G=1:H=0
394 IF INKEY$ ="="AND GLET G=0:L=L+2:GCURSOR L:GPRINT A$:IF L>104GOTO 410
400 GOTO 390
410 BEEP 3,25:CURSOR 0:PAUSE "BRAVO!!!":PAUSE :S=S+(100-A*100)/4
417 BEEP 1:TIME =TIME *8:A=TIME *100:CURSOR 3:PAUSE "TEMPS";A;" mn":PAUSE :GOSUB 6E4
420 WAIT 0:CLS :BEEP 2:PAUSE " *** SAUT A LA PERCHE ***":PAUSE :B$(0)="0163251F157341010101010101
421 WAIT :PRINT "(SPACE) Pour abaisser la":PRINT "Perche et (ENTER) Pour":PRINT "sauter...":WAIT 0
425 CLS :W=85:B$(1)="7800404040404040404040"
427 FOR J=1TO 3:CLS
430 GPRINT B$(0):GCURSOR 100:GPRINT B$(1)
435 GCURSOR 90:GPRINT 64:BEEP 1
440 FOR I=0TO 80:GCURSOR I:GPRINT 0;B$(0):IF INKEY$ =" "GOTO 450
443 NEXT I
445 BEEP 3,45:CURSOR 0:PAUSE "SPLAT !!";J:PAUSE :NEXT J:GOTO 500
450 FOR L=ITO I+10:GCURSOR L:GPRINT "6020101F1C6B4A0C081020":IF INKEY$ =CHR$ 13GOTO 460
452 NEXT L:GOTO 445
460 IF L<WAND L<100GOTO 445
461 GCURSOR 83:GPRINT "0000000000000000000000000000000000"
462 WAIT 40:GCURSOR 92:GPRINT "271A1C181C2B44067C":GCURSOR 92:GPRINT "00000001071977017F050909"
463 GCURSOR 93:GPRINT "00000000007C027A3F6A5A524040404040"
464 GCURSOR 95:GPRINT "030C3040007C0050507C727050686C"
465 GCURSOR 93:GPRINT "020408102040"
466 GCURSOR 90:GPRINT "101010202020404040":GCURSOR 89:GPRINT "404040404040404040"
467 WAIT 0:BEEP 1,18:CURSOR 0:PAUSE "BRAVO !!!":PAUSE :D=D+1:ON DGOTO 470,480,490,500
470 W=W+1:B$(1)="7C"+RIGHT$ (B$(1),LEN B$(1)-2):GOTO 427
480 W=W+1:B$(1)="7E"+RIGHT$ (B$(1),LEN B$(1)-2):GOTO 427
490 W=W+1:B$(1)="7F"+RIGHT$ (B$(1),LEN B$(1)-2):GOTO 427
500 CLS :S=S+D*5:GOSUB 6E4
510 A=0:CLS :BEEP 2:PAUSE " *** 110 METRES HAIES ***":PAUSE
512 WAIT :PRINT "(ENTER) Pour sauter":WAIT 0:CLS
515 CLS :CURSOR 5:INPUT "Vitesse (0-5)?";P:WAIT 5-P
520 CLS :FOR I=20TO 120STEP 20:GCURSOR I:GPRINT 112:NEXT I:GCURSOR 140:GPRINT 127
530 GCURSOR 0:GPRINT A$:GCURSOR 142:PAUSE "OK":PAUSE :BEEP 1:GCURSOR 142:PRINT "GO":TIME =0
540 FOR I=0TO 136STEP 2:GCURSOR I:GPRINT A$:IF INKEY$ =CHR$ 13GOTO 570
550 NEXT I:BEEP 2,31:GOTO 590
570 IF POINT (I+9)=112OR POINT (I+8)=112GCURSOR I+2:I=I+6:GPRINT "18090907770A3B23":GOTO 550
580 A=A+2:GOTO 550
590 A=A+TIME *1E4:FOR I=20TO 120STEP 20:IF POINT I<112LET A=A+10
595 NEXT I:PAUSE "TEMPS";A;" sec":PAUSE :S=S+(100-A)/5:GOSUB 6E4
600 WAIT 0:CLS :BEEP 2:PAUSE " *** LANCER DU DISQUE ***":CLS
610 WAIT :PRINT "(ENTER) Pour lacher le":PRINT "disQue...":WAIT 0:CLS
620 A$(0)="030304080000":A$(1)="0000031F0000":A$(2)="000008040303"
622 A$(3)="000008080C0C":A$(4)="000004083030":A$(5)="000078600000"
624 A$(6)="303008040000":A$(7)="0C0C08080000"
625 FOR J=1TO 3:WAIT 0:INPUT "Vitesse(0-3)?";U:CLS :FOR L=15TO 115STEP 5:GCURSOR L:GPRINT 127:NEXT L
630 GCURSOR 6:GPRINT 127:I=-1:WAIT 3-U
640 FOR I=0TO 7:GCURSOR 8:GPRINT A$(I):IF INKEY$ =CHR$ 13BEEP 1:GOTO 650
645 NEXT I:GOTO 640
647 GOTO 640
650 IF I<1OR I>4BEEP 2,32:CURSOR 20:PAUSE "SPLAT!":PAUSE :NEXT J:GOTO 670
652 IF I=1LET K=80
653 IF I=2LET K=40
654 IF I=3LET I=RND 2*4:K=20+(I=4)*10
655 IF I=4LET I=16:K=10
660 WAIT 2:K=K/((3-U)/2+1)+RND 5+20:FOR P=15TO K:GCURSOR P:GPRINT 0+127*(POINT P>I);POINT (P+1)OR I:NEXT P
665 CURSOR 20:PAUSE K-10;" m":PAUSE :NEXT J
670 IF K-10>XLET X=K-10
675 S=S+X/5:GOSUB 6E4:A$(0)="0163251F654509"
680 A$(1)="02066A2A1F694905":A$(2)="046C2C1F6A4A1101":A$(3)="086C241F6A4A1101"
685 A$(4)="201018146A5F3422"
690 CLS :BEEP 2:PAUSE "*** LANCER DU JAVELOT ***":PAUSE
695 WAIT :PRINT "(SPACE) Pour incliner le":PRINT "javelot et (ENTER) Pour le"
697 P=0:PRINT "lacher...":WAIT 0:CLS
700 FOR J=1TO 3:BEEP 1:R=0:Y=0:L=0:CLS :GCURSOR 100:GPRINT 64;64
710 FOR I=0TO 94STEP 2:IF INKEY$ =" "LET L=I:R=1
715 IF RAND INT ((I-L+1)/3)=(I-L+1)/3LET Y=Y+1:IF Y=5GOTO 735
717 GCURSOR I:GPRINT 0;A$(Y)
720 IF INKEY$ =CHR$ 13GOTO 750
730 NEXT I
735 BEEP 2,55:CURSOR 19:PRINT "SPLAT";J:PAUSE :NEXT J:GOTO 780
750 FOR K=100TO 155:GCURSOR K:GPRINT 0;2;2;2;2;2:NEXT K
752 IF Y=4LET P=I-85+RND 5
753 IF Y=0LET P=I-75+RND 5
754 IF Y=1LET P=I-55+RND 5
755 IF Y=3LET P=I-45+RND 5
756 IF Y=2LET P=I-40+RND 5
757 BEEP 2,22:CURSOR 0:PAUSE P;" m":PAUSE
760 IF P>XLET X=P
765 NEXT J
780 S=S+X/5:GOSUB 6E4
800 Q=0:WAIT 0:CLS :BEEP 2:PAUSE "    *** 1500 METRES ***":PAUSE
801 PRINT "(8)=";:GPRINT F$;:PRINT " (6)=";:GPRINT G$;:PRINT " (2)=";:GPRINT H$;:PRINT " (4)=";:WAIT :GPRINT I$
805 WAIT 0
810 CLS :GCURSOR 0:GPRINT "7F4141":GCURSOR 98:GPRINT "41417F"
812 FOR I=3TO 97:GCURSOR I:GPRINT 73:NEXT I:GCURSOR 110:GPRINT 127:GCURSOR 50:GPRINT &79:GCURSOR 51:GPRINT 16OR POINT 51
815 CURSOR 19:PAUSE "PRET!":FOR I=1TO 150:IF INKEY$ =""NEXT I:GOTO 819
817 BEEP 2,35:CLS :PAUSE "Vous etes Parti troP tot!":PAUSE :GOTO 810
819 BEEP 1:CURSOR 19:PRINT "GO!!!":TIME =0
820 I=51:K=16
830 IF INKEY$ ="8"LET K=K/2:IF K=1OR K=8LET K=K+K*(K=1OR I<98)
840 IF INKEY$ ="2"LET K=K+K:IF K=64OR K=8GOSUB 890
850 IF INKEY$ ="6"LET I=I+1:IF I=100LET I=I-1
860 IF INKEY$ ="4"LET I=I-1:IF I=0LET I=I+1
865 IF I=50GCURSOR 1:GPRINT "414149":Q=Q+1:GCURSOR 97:GPRINT "494141":CURSOR 19:PRINT Q*375;" m":IF Q=4BEEP 1:GOTO 900
870 GCURSOR I:IF I<3OR I>95GPRINT KOR POINT I:GOTO 830
880 GPRINT 73;K+73;73:GOTO 830
890 IF K=8AND I<3RETURN
895 K=K/2:RETURN
900 A=TIME *1E2:CURSOR 19:PAUSE A;" mn":PAUSE :S=S+(1.4-A)*60:GOSUB 6E4
1000 CLS :PRINT "*** SCORE TOTAL="+STR$ INT (S*50)+" ***"
1010 WAIT :PRINT :END
60000 CLS :PAUSE "SCORE=";INT (S*50);" Points":PAUSE :D=D-25:CLS
60002 IF PEEK &B002=77GLCURSOR (165,D):LPRINT STR$ INT (S*50):GLCURSOR (0,-330)
60005 RETURN
61000 CLEAR :DIM A$(0)*26:A$(0)="*) DECATHLON OLYMPIQUE (*"
61010 WAIT 0:FOR I=1TO 25:FOR J=25TO ISTEP -1:CURSOR J:PRINT MID$ (A$(0),I,1):NEXT J:BEEP 1,25-I:NEXT I
61021 CLEAR :F$="000004027F0204":G$="00080808082A1C08"
61022 H$="000010207F2010":I$="00081C2A08080808"
61029 IF PEEK &B002<>77PAUSE :PAUSE :RETURN
61030 TEXT :COLOR 1:CSIZE 3:LPRINT "***DECATHLONOLYMPIQUE***":LF 3:GRAPH :SORGN
61040 LINE (0,0)-(215,-250),,2,B:LINE (160,0)-(160,-250)
61042 LINE (20,-250)-(20,0)
61045 FOR I=-25TO -225STEP -25:LINE (0,I)-(215,I):NEXT I
61050 BEEP 1:INPUT "PRENOM:";N$:GLCURSOR (185,85):ROTATE 1:COLOR 0:LPRINT LEFT$ (N$,8)
61060 ROTATE 0:FOR I=-25TO -225STEP -25:GLCURSOR (5,I+5):LPRINT STR$ (-I/25):NEXT I:GLCURSOR (0,-245):LPRINT "10":CSIZE 1
61070 GLCURSOR (25,-21):COLOR 3:LPRINT "100 Metres":GLCURSOR (25,-46):LPRINT "Saut en hauteur":GLCURSOR (25,-71)
61080 LPRINT "Lancer du Poids":GLCURSOR (25,-96):LPRINT "Saut en lon9ueur":GLCURSOR (25,-121)
61090 LPRINT "400 Metres":GLCURSOR (25,-146):LPRINT "Saut a la Perche":GLCURSOR (25,-171)
61100 LPRINT "110 Metres Haies":GLCURSOR (25,-196):LPRINT "Lancer du dis9ue":GLCURSOR (25,-221)
61110 LPRINT "Lancer du javelot":GLCURSOR (25,-246):LPRINT "1500 Metres":CSIZE 2:COLOR 1:RETURN
```




# CALCULS FINANCIERS

**Ce programme calcule simultanément les 5 variables financières N, I, PV, PMT, FV.**

Ces variables sont affectées, en DEF, aux touches « A » (N), « S » (I), « D » (PV), « F » (PMT) et «G » (FV). Il est conseillé de se servir d'un normographe (gabarit). Les lettres A, S, D, F, G contiennent également les valeurs non arrondies de ces variables, l'arrondi ne se faisant qu'à l'affichage.

## MODE D'EMPLOI

Entrer une donnée au clavier en mode DEF puis taper SHIFT + la touche correspondant à la fonction demandée.
Exemple : 7,5% annuel : Taper 7.5/12 SHIFT S
Si vous tapez SHIFT TOUCHE, écran vide (CL), vous obtenez, à l'affichage, la valeur actuelle de la donnée. Autre méthode : Variable (lettre correspondante) ENTER.
Exemple : pour N : A ENTER
Pour effacer tout : CLEAR ; partiellement : A=0 ENTER S=0 ENTER etc.
Les crédits se rentrent en valeurs positives, les débits, en valeurs négatives (précédées du signe « − »). Le calcul de chaque variable se lance par SHIFT TOUCHE de la sus-dite variable ; affichage vide (CL).

## CONTENU DES VARIABLES

| | |
|---|---|
| A | N |
| B | I/100 |
| C | 1+I/100 |
| D | PV |
| F | PMT |
| G | FV |
| H | (1+I/100) ↑ (−N) |
| I | Entrée de I, comparaison avec S |
| J | PMT/(I/100) |
| K | entrée de FV, comparaison avec G |
| L | Entrée de PMT, comparaison avec F |
| M | Entrée de PV, comparaison avec D |
| N | Entrée de N, comparaison avec A |
| O | Arrondi affichage |
| S | I |
| Z | (1−(1+I/100) ↑ (−N))/(I/100) |

```
  1:REM "CALCULS
     FINANCIERS"
  2:REM *N,I,PV,
     PMT,FV*
  3:REM JFV
 10:"A"AREAD N:
     IF (N<>A)*(N
     <>0)LET A=N:
     END
 15:J=F/B:A=LN (
     (J-G)/(J+D))
     /LN (B+1):A=
     INT (INT (10
     A+5)/10):
     PRINT "N=";A
     :END
 20:"S"AREAD I:B
     =I/E2:C=B+1:
     IF (I<>S)*(I
     <>0)LET S=I:
     END
 25:IF F<>0GOTO
     35
 30:B=(-G/D)^(1/
     A)-1:GOTO 45
 35:Y=(ABS G-ABS
     (FA+D))*(ABS
     (FA+G)-ABS D
     ):B=E-9Y/ABS
     Y
 40:GOSUB E2:E=H
     A/C:W=(ZF+D+
     GH)/((EF-ZF)
     /B-GE):B=B-W
     :W=ABS W:IF
     E-7<=WGOTO 4
     0
 45:S=E2B:O=S:
     GOSUB 110:
     PRINT "I=";O
     ;" %":END
 50:"D"AREAD M:
     IF (M<>D)*(M
     <>0)LET D=M:
     END
 55:GOSUB E2:D=-
     FZ-GH:O=D:
     GOSUB 110:
     PRINT "PV=";
     O;" FR":END
 60:"F"AREAD L:
     IF (L<>F)*(L
     <>0)LET F=L:
     END
 65:GOSUB E2:F=-
     (D+GH)/Z:O=F
     :GOSUB 110:
     PRINT "PMT="
     ;O;" FR":END
 70:"G"AREAD K:
     IF (K<>G)*(K
     <>0)LET G=K:
     END
 75:GOSUB E2:G=-
     (ZF+D)/H:O=G
     :GOSUB 110:
     PRINT "FV=";
     O;" FR":END
100:C=B+1:H=C^(-
     A):Z=(1-H)/B
     :RETURN
110:IF OLET O=
     INT (E2O+.5)
     /E2
120:IF -OLET O=-
     INT (-E2O+.5
     )/E2
130:RETURN
```

# AWELE

L'Awele est joué par les Baule de la Côte d'Ivoire. Le vrai nom est ALE. Il fait partie des Waris africains.

Disposition du jeu : le tablier (l'écran) comporte deux camps de 5 cases chacun. Le camp gauche est celui du 1251.

On joue avec 30 graines ou coquillages... au début, chaque case en reçoit trois.

Au chaque coup, qui comporte un seul semis, le joueur prend le contenu d'une case non vide de son camp et sème les graines une par une dans les cases suivantes. De gauche à droite.

Il suffit ici d'indiquer au 1251 le numéro de la case que l'on désire jouer. Les cases sont numérotées de 1 à 10. Les cases 6, 7, 8, 9 et 10 constituent, sur la partie droite de l'écran, le camp du joueur.

**Exemple d'un coup :**

avant 2 3 0 1 3 / 1 2 1 3 1
case ? 9
après 3 4 0 1 3 / 1 2 1 0 2

Si la dernière graine semée tombe dans une case adverse contenant une graine, le joueur prend le contenu de cette case d'arrivée, y compris la dernière graine qu'il vient d'y déposer. Il prend également le contenu des cases adverses situées avant, si elles contiennent au plus deux graines à l'issue du coup. Les cases permettant une prise doivent former une chaîne ininterrompue. Ces prises constituent le gain du joueur.

**Exemple d'une prise :**

avant 1 2 1 1 0 / 1 2 0 6 4
case ? 10
/nb/ : la case 9 ne réalise pas une prise
après coup 2 3 2 2 0 / 1 2 0 5 0
/nb/ : prise des cases 3 et 4
après prise 2 3 0 0 0 / 1 2 0 5 0

Bien que ce jeu consiste à acquérir le plus de graines possible, la règle « donner à manger » oblige à ne pas priver son adversaire de graines. Cette règle est ici interprétée comme suit : la privation est pénalisée si elle se réalise sur un coup-prise.

**Exemple :**

avant 1 1 0 0 0 / 0 0 0 3 1
case ? 9
après coup 2 2 0 0 0 / 0 0 0 0 2
après prise 0 0 0 0 0 / 0 0 0 0 2

Le jeu s'arrête sur cette prise illégale, et le score du fauteur est diminué du montant de sa prise (– 4). Dans les autres cas, le joueur sans graine passe son tour (le 1251, bien entendu, effectue lui-même ces contrôles).

Le jeu s'arrête lorsque l'un des deux joueurs a pris au moins 15 graines, ou lorsque le total des gains des deux joueurs dépasse 21 graines.

A tout moment le joueur peut consulter son score par la réponse 0 à la question : case ?

L'Awele nocturne étant réservé aux divinités, les hommes ne peuvent jouer que le jour. Si une femme veut mettre au monde un garçon, elle doit jouer avec un garçon...

Jack Vimond

## EXPLICATIONS LIGNE A LIGNE :

| | | |
|---|---|---|
| 1 à 3 | | Initialisation : 3 graines par case |
| 4 | | Tirage au sort du premier à jouer |
| 10 à 15 | | Entrée numérique du coup du joueur, test si passe tour, si 0 affichage score |
| 20 à 50 | | Tri du coup, semis de gauche à droite |
| 60 à 64 | | Test de la prise |
| 70 à 84 | | Effectue la prise |
| 85 à 86 | | Pénalisation pour non respect de la règle « Donner à manger » |
| 90 | | Affichage |
| 95 | | Arrêt du jeu : si un joueur a 15 graines ou si le total des graines dépasse 21 |
| 100 à 106 | | A qui de jouer ? |
| 200 | | Passage du tour |
| 201 à 214 | | Test si prise ou perte, et test règle « D à M » |
| 302 à 316 | | TEST si n° case + contenu supérieur à 9 |
| | STRATEGIE | Test si case = 1 |
| 400 | DU PC | Si 1er tour jeu aléatoire |
| 500 à 512 | | Test cases dans l'ordre croissant |
| 600 à 634 | | Double tableau, avance et test être pris |
| 700 à 716 | | Test risque si avance case contenant 1 graine |
| | S/P DE | |
| 800 à 824 | SIMULATION | Double tableau, avance et décompte perte |
| 850 à 870 | | Test règle « D à M » avant prise |
| 900 | FIN | Affichage score |

```
  1:CLEAR : DIM B(10),C(
    10)
  2:FOR I=1 TO 10:B(I)=3
    : NEXT I
  3:PRINT B(1);B(2);B(3)
    ;B(4);B(5);"/";B(6);
    B(7);B(8);B(9);B(10)
  4:IF ( RND 2)=2 THEN "
    TOU"
 10:"JOU"
 11:IF B(6)=0 AND B(7)=0
    AND B(8)=0 AND B(9)=
    0 AND B(10)=0 BEEP 1
    : GOTO "TOU"
 13:INPUT "CASE ? ";I
 14:IF I=0 THEN 900
 15:IF B(I)=0 OR I<6 OR
    I>10 THEN 11
 20:"TRI"B(0)=B(I):C=0
 22:IF B(I)+I<=10 LET A=
    I+1:B=B(I)+I: GOTO "
    AV1"
 24:IF I=10 AND B(I)<=10
    LET A=1:B=B(I): GOTO
    "AV1"
 26:IF I=10 AND B(I)>10
    LET A=1:B=10:C=1:
    GOSUB "AV1":A=1:B=B(
    0)-10: GOTO "AV2"
 28:IF B(I)+I>10 LET A=I
    +1:B=10:C=1: GOSUB "
    AV1"
```

```
29:IF B(0)-10+I<=10 LET
    A=1:B=B(0)-10+I:
    GOTO "AV2"
30:IF B(0)-10+I<=20 LET
    A=1:B=10:C=1: GOSUB
    "AV2":A=1:B=B(0)-20+
    I: GOTO "AV2"
40:"AV1"B(I)=0
41:"AV2"
42:FOR N=A TO B
44:B(N)=B(N)+1
46:NEXT N
48:IF C=1 LET C=0:
    RETURN
50:PRINT B(1);B(2);B(3)
    ;B(4);B(5);"/";B(6);
    B(7);B(8);B(9);B(10)
60:"TPR"
62:IF B(N)<>2 THEN "TOU
    "
63:IF B(N)=2 AND N<6
    AND I<6 THEN "TOU"
64:IF B(N)=2 AND N>5
    AND I>5 THEN "TOU"
70:"PRI"
71:Z=0
72:FOR N=B TO S STEP -1
74:IF B(N)>2 LET N=S:
    GOTO 84
76:IF S=6 LET Q=Q+B(N):
    Z=Z+B(N)
78:IF S=1 LET W=W+B(N):
    Z=Z+B(N)
82:B(N)=0
84:NEXT N
85:IF B(6)=0 AND B(7)=0
    AND B(8)=0 AND B(9)=
    0 AND B(10)=0 LET Q=
    Q-Z: BEEP 1: GOTO 90
    0
86:IF B(1)=0 AND B(2)=0
    AND B(3)=0 AND B(4)=
    0 AND B(5)=0 LET W=W
    -Z: BEEP 1: GOTO 900
90:PRINT B(1);B(2);B(3)
    ;B(4);B(5);"/";B(6);
    B(7);B(8);B(9);B(10)
95:IF Q>14 OR W>14 OR Q
    +W>21 BEEP 2: GOTO 9
    00
100:"TOU"
102:E=E+1
103:S=(E/2- INT (E/2))
104:IF S<>.5 LET S=1:
    GOTO "JOU"
106:S=6
200:IF B(1)=0 AND B(2)=0
    AND B(3)=0 AND B(4)=
    0 AND B(5)=0 BEEP 1:
    GOTO "TOU"
201:PAUSE "ORDINATEUR"
202:D=0:G=0
204:FOR O=6 TO 10
206:FOR P=5 TO 1 STEP -1
208:IF B(O)=1 AND B(P)+P
    =O LET G=P
210:IF B(P)=1 AND B(O)+O
    -10=P LET D=P
212:NEXT P: NEXT O
213:IF G<>0 THEN "SI0"
214:IF D<>0 LET I=D:
    GOTO "TRI"
302:K=0:L=0:M=0
304:FOR P=1 TO 5
306:IF B(P)+P>9 LET M=P
308:IF B(P)=1 AND K<>0
    LET L=P
310:IF B(P)=1 AND K=0
    LET K=P
312:NEXT P
313:IF M<>0 LET I=M:
    GOTO "TRI"
314:IF K<>0 AND B(K+1)=0
    AND K<>5 THEN "SI2"
316:IF K<>0 LET I=K:
    GOTO "TRI"
400:IF E=1 LET I= RND 3:
    GOTO "TRI"
500:"DPN"S=0
502:FOR K=1 TO 5
504:IF B(K)<>0 THEN "SI1
    "
505:IF B(K)=0 LET S=S+1
506:NEXT K
508:IF V=0 THEN "TRI"
510:IF V<>0 AND U=0 AND
    S<4 THEN "TRI"
511:IF V<>0 AND U=0 AND
    S=4 LET I=Y: GOTO "T
    RI"
512:IF V<>0 AND U<>0
    THEN "SI3"
550:REM COPYRIGHT VIMOND
     JACK
600:"SI1"
602:FOR I=1 TO 10
604:C(I)=B(I)
606:NEXT I
610:C(K)=0
612:FOR I=K+1 TO B(K)+K
614:C(I)=C(I)+1
616:NEXT I
622:FOR I=1 TO 5
624:FOR J=6 TO 10
626:IF C(I)=1 AND C(J)+J
    -10=I AND V=0 LET V=
    I:Y=K
627:IF C(I)=1 AND C(J)+J
    -10=I AND V<>0 AND U
    =0 AND K>Y LET U=I:X
    =K
628:NEXT J
630:NEXT I
632:IF V=0 LET I=K:K=5:
    GOTO 506
633:IF V<>0 AND U=0 AND
    K<>Y LET I=K:K=5:
    GOTO 506
634:GOTO 506
700:"SI2"
702:R=0
704:FOR J=10 TO 6 STEP -
    1
706:IF K+1=B(J)+J-10 LET
    J=6:R=1
708:NEXT J
710:IF R=0 LET I=K: GOTO
    "TRI"
712:IF L<>0 AND L<>5 AND
    B(L+1)=0 LET K=L:
    GOTO 702
714:IF L<>0 LET I=L:
    GOTO "TRI"
716:GOTO "DPN"
800:"SI3"
802:FOR I=1 TO 10
804:C(I)=B(I)
806:NEXT I
810:Z=V
812:FOR I=Z TO 1 STEP -1
814:C(I)=C(I)+1
816:IF C(I)>0 AND C(I)<3
    LET G=G+C(I)
818:NEXT I
820:IF H=0 LET V=U:H=G:G
    =0: GOTO 802
822:IF H<=G LET I=Y:
    GOTO "TRI"
824:I=X: GOTO "TRI"
850:"SI0"
852:FOR I=5 TO 10
854:C(I)=B(I)
856:IF I<G+B(G) LET C(I)
    =C(I)+1
858:NEXT I
860:FOR I=G+B(G) TO 5
    STEP -1
862:IF C(I)>2 LET I=5:
    GOTO 866
864:C(I)=0
866:NEXT I
868:IF C(6)=0 AND C(7)=0
    AND C(8)=0 AND C(9)=
    0 AND C(10)=0 THEN 2
    14
870:I=G: GOTO "TRI"
900:PRINT "SCORE : TOI "
    ;W;" ORDI. ";Q
901:IF I=0 PRINT B(1);B(
    2);B(3);B(4);B(5);"/
    ";B(6);B(7);B(8);B(9
    );B(10): GOTO 13
902:END
```

# UN PEU DE LM DANS BEAUCOUP DE BASIC

**Même si vous n'êtes pas un fanatique du LM, vous pouvez néanmoins profiter de quelques découvertes des maniaques qui y engloutissent leurs nuits... Voici donc quelques astuces simples et inédites qui vont étendre les possibilités déjà grandes du Basic de votre PC 1251. Ces astuces ne demandent qu'à être découvertes sur d'autres modèles.**

## VISUALISATION A L'ECRAN D'UNE CHAINE DE PLUS DE 24 CARACTERES

Vous avez sans doute remarqué que l'on ne pouvait pas afficher directement plus de 24 caractères sur le LCD du PC 1251, hormis avec un programme LM donné dans le dernier N°.
On peut aussi le faire sans ce programme et plus simplement (sans toutes les possibilités offertes par le pgm LM), en intercalant la séquence POKE &C6DA,32 avant le PRINT expression de plus de 24 car.
La machine se met alors en mode édition comme si l'on venait de taper au clavier le texte visualisé, et on a la faculté de se balader avec le curseur et de lire la suite de la chaîne avec les touches :



Par contre, comme on devait s'y attendre, si on tape ENTER la ligne va se trouver interprétée et donnera le plus souvent un message ERROR N°. Il faut donc relancer le pgm par un RUN, GOTO, ou DEF étiquette. On peut cependant reprendre le pgm en séquence derrière le PRINT par ENTER ou CL et ENTER si ERROR. En outre si l'on reprend la suite du pgm par DEF key et que l'instruction exécutée est un AREAD var, la chaîne affichée (par ex. une var B$(0)), qui a pu être modifiée par l'éditeur-touches INS et DEL est ressaisie. Il s'agit de l'émulation de l'instruction LINPUT de certains BASICs.

## SORTIE PREMATUREE ET CAVALIERE D'UNE BOUCLE FOR-NEXT :

Contrairement à d'autres langages (par ex. FORTRAN), peu de BASICs admettent que l'on sorte d'une boucle autrement qu'avec le NEXT et la valeur saturée de l'indice de la boucle. Si on essaie de sortir par un GOTO pointant à l'extérieur de ladite boucle, le niveau va rester « empilé » dans le tampon (cf. articles dans les N° 8 et 10). Ce qui entraîne souvent un message d'erreur suite au dépassement du maximum d'imbrications autorisé.
Certains BASICs possèdent une instruction (POP, EXIT ou autre) permettant de « dépiler » le niveau de boucle avant d'effectuer la sortie cavalière par GOTO, mais pas le PC 1251.

La solution classique est la suivante :

```
10  FOR I=1 TO max
       ...
       ...
       IF (condition quitter boucle, vraie) LET
         I=max:GOTO 100
       ...
       ...
100 NEXT I
```

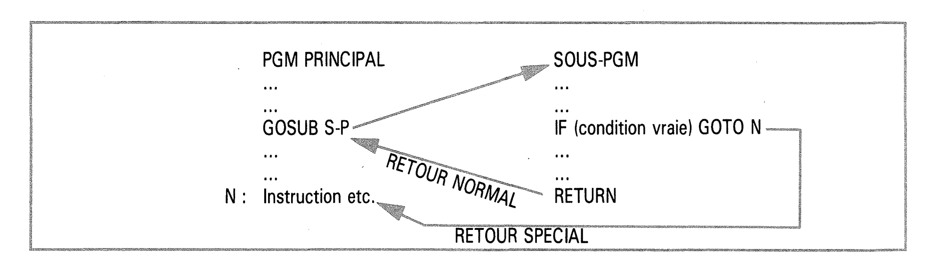
Ceci est un exemple de solution qui ne convient pas si ;
- le NEXT n'est pas en tête de ligne.
- on veut aller ailleurs qu'à la suite du NEXT.

On peut remplacer la fonction POP par : POKE &C6F3,PEEK &C6F3-&10 avant le GOTO audacieux (ou plus court : X=&C6F3: POKE X,PEEK(X-16).

Voir le pgm 1 de démonstration où l'on veut sortir de la boucle comptant de 1 à 5, à la valeur 3. Essayer d'ôter le POKE en ligne 520, vous obtiendrez une ERROR 5.

## SORTIE D'UN SOUS-PROGRAMME AVEC RETOUR AILLEURS QU'A L'APPEL

Il s'agit d'une très mauvaise pratique de programmation : on veut lors de certaines conditions, sortir du sous-pgm non par le RETURN classique renvoyant à la sortie du GOSUB appelant, mais par un GOTO adr. Cependant, cela peut permettre dans l'environnement restreint des 3486 octets du PC 1251, de gagner de précieux octets ou de « simplifier » un algorithme. Ici encore, comme avec le FOR-NEXT, il faut « dépiler » le niveau de sous-pgm empilé lors du GOSUB :




On peut faire cela avec :
POKE &C6F4,PEEK &C6F4–4
Dans l'exemple du pgm 2, si on enlève le POKE : ERROR 5

## EXTENSION DE LA FONCTION VAL

Rappelons que la fonction VAL dont l'argument est une chaîne ou variable alphanumérique donne comme résultat la valeur numérique, si elle existe et si elle figure en tête des caractères, correspondant à la chaîne argument interprétée comme du numérique.

**Par exemple :**
VAL"123.5" donne le nombre 123.5
A$="123.5":B=VAL A$ charge dans la var la valeur 123.5, et l'on peut écrire C=1+VAL A$ qui donne 124.5 etc.
L'intérêt est, par exemple, de pouvoir stocker du numérique dans des var. alpha (ou tableau) de façon plus concise. Ainsi pour un fichier de

nombre de 3 chiffres max. (entiers inférieurs à 1000), on aura besoin de 3 octets par nombre (plus 7 de déclarations de tableau) en utilisant le tableau B$(x)*3 pour les contenir, alors que des variables numériques nécessitent 8 octets. On récupère ensuite les valeurs grâce à VAL.

Par contre certains micros ont une fonction VAL plus évoluée, permettant de fournir la valeur numérique résultante du calcul d'une expression valide contenue dans une chaîne de caractères. Par exemple VAL "1+1" donne 2 (le 1251 répond 1 dans ce cas) et VAL "(-B+SQR(B*B-4*A*C))/2*A" donne 2 si A=1,B=-3,C=2, etc.

Il est à ce moment possible de créer un fichier alpha contenant de véritables formules. Ou bien dans un pgm traceur de graphe de fonction, on peut demander l'entrée de l'équation par un INPUT B$(0).

L'astuce révélée ci-après permet d'émuler une telle pseudo-fonction VAL évoluée.

Réalisation :

- Dimensionner avant tout autre tableau une variable B$(0)*80
- Intégrer le sous-pgm 3

**NB** : le N° de ligne et l'étiquette n'ont aucune importance.

- Il vous suffit alors de placer comme vous voulez une formule dans B$(0), puis de faire GOSUB "VAL" (dans le cas du pgm 3) pour récupérer la valeur numérique associée dans la variable A (équivaut à A=VALe BS(0)), VALe étant la pseudo-fonction.

La formule peut comporter les signes +, −, *, /, &, .,.E, , (, et ;) ainsi que toute variable numérique telle que A,B,Z,C(1), etc.

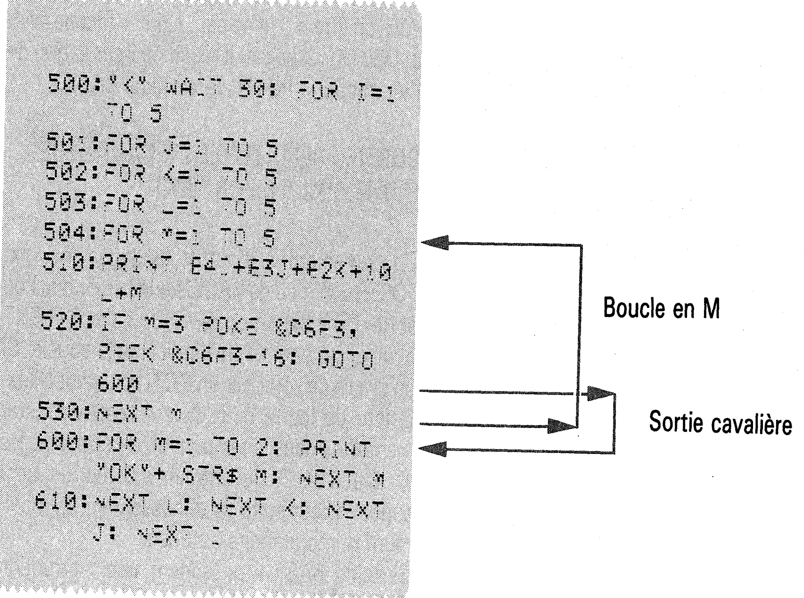
Mais une fonction composée de lettres comme SIN,COS, etc. ne peut directement être utilisée. Les amateurs de LM pourront résoudre ce problème. La longueur maximale de l'équation est de 80 caractères.

**NB** : il est concis et plus rapide de réduire le sous-pgm à : "VAL" CALL 51104:RETURN et de placer le POKE générateur du LM à l'extérieur (il suffit de le réaliser une fois), plutôt que de laisser le POKE dans le s-p.

Vous pouvez tester le pgm 4 de test en faisant :

| opération | résultat |
|---|---|
| 1+1 | 2 |
| 1.25E−2*10 2+B | 6.25 si B=5 |
| 2*(Q=0)+3*(Q=1) | 2 si Q=0 et 3 si Q=1 |

François BOESSER



```
500:"<" WAIT 30: FOR I=1
     TO 5
501:FOR J=1 TO 5
502:FOR K=1 TO 5
503:FOR L=1 TO 5
504:FOR M=1 TO 5
510:PRINT E4I+E3J+E2K+10
     L+M
520:IF M=3 POKE &C6F3,
     PEEK &C6F3-16: GOTO
     600
530:NEXT M
600:FOR M=1 TO 2: PRINT
     "OK"+ STR$ M: NEXT M
610:NEXT L: NEXT K: NEXT
     J: NEXT I
```



pgm 1

```
500:"<" INPUT "NORMAL OU
     SPECIAL(N/S)?";A$
501:GOSUB "SP1"
502:PRINT "RETOUR NORMAL
     ": END
503:"SPECIAL" GOSUB "*":
     PRINT "RETOUR SPECIA
     L": END
601:"SP1" GOSUB "SP2":
     RETURN
602:"SP2" GOSUB "SP3":
     RETURN
603:"SP3" GOSUB "SP4":
     RETURN
604:"SP4" GOSUB "SP5":
     RETURN
605:"SP5" GOSUB "SP6":
     RETURN
606:"SP6" GOSUB "SP7":
     RETURN
607:"SP7" GOSUB "SP8":
     RETURN
608:"SP8" GOSUB "SP9":
     RETURN
609:"SP9" GOSUB "SP10":
     RETURN
610:"SP10" IF A$="S"
     POKE &C6F4, PEEK &C6
     F4-4: GOTO "SPECIAL"
620:RETURN
630:"*" RETURN
```

pgm 2

```
110:"VAL" POKE &C7A0,&03
     ,&C5,&02,&80,&F1,&F1
     ,&F2,&05,&00,&07,&10
     ,&C6,&98,&A0,&19,&37
111:CALL &C7A0: RETURN
```

pgm 3

```
  1:DIM B$(0)*80
  2:POKE &C7A0,&03,&C5,&
     02,&80,&F1,&F1,&F2,&
     05,&00,&07,&10,&C6,&
     98,&A0,&19,&37
 10:"<" INPUT B$(0):
     GOSUB "VAL": PRINT A
     : GOTO "<"
110:"VAL" CALL &C7A0:
     RETURN
```

pgm 4




# SCRAMBLE

**Ce jeu intergalactique crée par PISIOX, est le plus fantastique jamais programmé sur un PC 1251/55 (c'est aussi le plus long).**

Il allie des possibilités sonores et graphiques uniquement possibles en LM. De plus, il exige beaucoup de réflexes de la part du joueur ; en un mot il est vraiment génial.

Du fait de sa longueur inhabituelle et qu'il est extrait d'un manuel japonais (I/O), le pgm n'est pas placé à une adresse permettant de le rentrer, ou de le désassembler facilement avec le pgm DESASSEMBLEUR du N° 8. Pour remédier au premier inconvénient, il suffit d'épurer le pgm du N° 8 comme suit. Vous effacez toutes les lignes sauf les lignes 1, 31, 32, 33, 34, 35. N'oubliez-pas de garder dans la RSV le pgm intitulé : PROG 1. De plus supprimer la fin de la ligne 32 à savoir le test : IF I>=&C1D0 END.

Comme tout SCRAMBLE classique, il s'agit de parcourir des tableaux sans se faire bètement tuer. Sur celui-ci, quatre-tableaux sont disponibles. Dans le premier vous êtes représenté par un trait continu de 5 points et vous avencez vers une caverne (ou plutôt elle avance vers vous) au rythme d'un accompagnement musical. Sept touches manœuvrent le vaisseau dans toutes les directions (voir ci-dessous).

Dans le deuxième tableau, vous n'avez que trois touches, et vous devez faire face à une horde d'OVNI ennemis dans la plus pure tradition SPACE INVADERS. Les deux autres tableaux doivent être dans le style du N° 1 puisqu'ils utilisent les sept mêmes touches. Mais, je ne peux vous en dévoiler davantage, n'ayant pas réussi à passer plus de deux tableaux. Riez, vous verrez, c'est pas si facile !! Bonne chance.

J.F.V.

**Tableau 1, 3, 4**

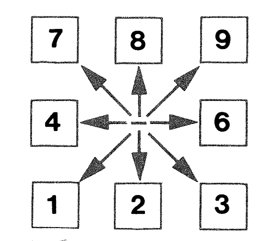



**Tableau 2**

| | |
|---|---|
| 8 | Montée |
| 5 | Tir |
| 2 | descente |

```
DUMP SCRAMBLE
(C) PISIOX I/O

BD00 783F 3010 3F5E 841A
BD08 6742 280B 245A 6780
BD10 2811 0201 2C0D 6748
BD18 2814 24D2 6700 2803
BD20 0240 5205 5205 5205
BD28 5205 522C 6567 4528
BD30 6103 0224 103D 4352
BD38 1147 5211 4B52 114F
BD40 5204 D508 0424 6708
BD48 3810 D508 0404 D508
BD50 C2DA 670A DA38 3B2D
BD58 17D2 10BD 6352 1167
BD60 5205 D510 0404 D510
BD68 DA03 80F1 EE43 F1E9
BD70 2527 6700 2905 0231
BD78 FF31 0200 4267 0029
BD80 04FF 3110 8009 5774
BD88 0252 67A8 2804 79BE
BD90 A810 800A 5742 5267
BD98 0928 2902 0052 0380
BDA0 F1F5 42F1 F124 26C2
BDA8 DA67 88DA 2908 783F
BDB0 2067 0028 0242 88DB
BDB8 0200 7403 4929 0410
BDC0 8008 5210 F80A 0200
BDC8 0031 1F02 0103 80F1
BDD0 F102 0303 F8F1 F588
BDD8 0208 DB24 FACF 7400
BDE0 03BE F1F1 2426 2426
BDE8 2426 0606 FACF 4929
BDF0 1510 BDDF 5774 0067
BDF8 7828 0302 0052 2C91
BE00 0000 0002 0102 0402
BE08 0408 0408 1008 1020
BE10 1020 4020 4000 4000
BE18 0000 0008 0408 1008
BE20 1020 1020 4020 4000
BE28 4000 1008 1020 1020
BE30 0000 0020 1020 4020
BE38 4000 4000 1008 1020
BE40 1020 0402 0408 0408
BE48 0000 0000 4000 1008
BE50 1020 1020 0402 0408
BE58 0408 1008 1002 0102
BE60 0000 0020 1020 0402
BE68 0408 0408 1008 1002
BE70 0102 0402 0408 0408
BE78 0000 0000 0000 0000
BE80 1002 0102 040A 0408
BE88 0408 1028 1020 1020
BE90 1080 0057 6700 2802
BE98 3778 1D11 10C6 9257
BEA0 4252 3700 0000 0000
BEA8 10C6 9257 7402 5237
BEB0 10FF AA57 4252 D620
BEB8 0200 2803 0218 10BD
BEC0 F652 78BD 0010 8009
BEC8 57D2 D210 BED5 5210
BED0 FF50 0200 002A 1F37
BED8 78BE ED78 11E0 78BE
BEE0 F310 F83F 5742 5267
BEE8 007C BED8 3712 5F61
BEF0 52DF 3710 4000 0200
BEF8 002A 1F37 0000 0000
BF00 10C6 E557 7480 03BF
BF08 F1F1 2410 BEF9 5224
BF10 10F8 3F52 79BE D800
BF18 10B8 0002 0000 2F1F
BF20 7840 1810 FFB4 D407
BF28 5737 0000 0000 0000
BF30 73BF 60FF 4410 C6E5
BF38 5237 0000 0000 0000
BF40 78BE B010 C692 5767
BF48 5138 0237 7811 E079
BF50 BF40 0000 0000 0000
BF58 0000 0000 0000 08F8
BF60 02FD 422A 0302 F010
BF68 BF61 5275 B003 C5F1
BF70 F110 C433 5734 02D2
BF78 5224 78C4 185B 10C4
BF80 3352 3700 0000 0000
C000 7747 4757 5750 5050
C008 1010 5020 282E 2E2E
C010 2E20 2030 2021 2720
C018 2020 2020 2036 3232
C020 2424 0909 1313 4747
C028 4747 5353 5858 5C5C
C030 5C5C 5C4C 4060 7079
C038 7D41 4141 4141 5F41
C040 0000 0000 0802 0208
C048 0002 2004 0040 0802
C050 0820 0104 0110 0040
C058 4141 4141 4141 7163
C060 6167 6367 6F67 6767
C068 6767 6767 6373 7179
C070 797D 7D41 4141 4141
C078 4141 4141 5555 5557
C080 4141 4141 415D 5D41
C088 4141 4141 4141 5541
```

```
C090 5541 5541 5541 5541
C098 5541 5541 7541 4141
C0A0 4141 4141 5541 5541
C0A8 5541 5541 5541 5541
C0B0 5541 5D41 4141 4141
C0B8 4141 5541 5541 5541
C0C0 5541 5541 5541 5541
C0C8 5541 5741 4141 4141
C0D0 4141 4549 4953 4541
C0D8 6151 4949 4141 4141
C0E0 4147 414F 4141 4171
C0E8 7141 4141 4141 4141
C0F0 4141 414F 4F41 4141
C0F8 4141 417D 4141 4100
C100 4761 7379 7870 7163
C108 6367 4341 4908 1C0C
C110 0646 6444 040C 1808
C118 0141 7341 6341 4147
C120 474F 4F4F 5F1F 0F4F
C128 4767 6341 4000 0808
C130 1C1C 3E3E 0800 0022
C138 2200 0049 4900 0022
C140 2200 0014 1400 0022
C148 2200 0049 4900 0022
C150 2200 0014 1400 0000
C158 0008 080F 0F00 0000
C160 7C78 0000 0E1E 0000
C168 0079 7343 0160 7078
C170 7C70 6043 0300 000E
C178 3E3C 1C0C 0464 6404
C180 044C 4800 0113 1341
C188 4000 0000 2424 243D
C190 1911 0300 0000 1F02
C198 0100 007C 5454 5C00
C1A0 0000 1715 151D 0000
C1A8 0002 021E 0202 5070
C1B0 0002 464C 0000 0077
C1B8 7700 0000 0073 7B00
C1C0 0000 005F 5F00 0000
C1C8 005B 5B00 0020 227E
C1D0 7E60 4040 060E 0E1C
C1D8 1C09 0948 480E 0E00
C1E0 0001 6777 7100 0000
C1E8 082A 2A08 0000 222A
C1F0 2A42 0000 0037 3703
C1F8 0141 6179 7161 4141
C300 78C4 90FF 443A 0510
C308 C691 4D34 0205 03F8
C310 F1E9 F1F5 0236 3424
C318 262F 0310 C696 821A
C320 4228 1510 F87B 5742
C328 5202 80D6 3F28 1910
C330 C690 D400 5B37 37F1
C338 F124 8363 802A 0478
C340 C4C8 10C6 9684 1B10
C348 F83B 5210 C692 0003
C350 8218 8302 0434 0524
C358 DA46 DA52 2F06 10C6
C360 9282 18DA 5B34 7A0A
C368 C3C0 6941 C3B8 42C3
C370 8F43 C3A3 44C3 9846
C378 C3A7 47C3 9748 C333
C380 49C3 AF67 C3BE 07C3
```

```
C388 B8C3 3E7A 01C3 9CDA
C390 6740 3803 D15A 377A
C398 07C3 84DA 8463 0438
C3A0 0205 377A 03C3 90DA
C3A8 8463 3638 0204 377A
C3B0 09C3 A8DA 6701 3803
C3B8 D1D2 3758 5B5B DA37
C3C0 03FF 8345 10C6 9282
C3C8 1911 9282 18DA 10C6
C3D0 915B 5202 0334 8324
C3D8 DA47 DA52 2F06 24C6
C3E0 3819 10F8 7B83 5510
C3E8 C694 5744 1196 5744
C3F0 5975 3728 0242 1190
C3F8 5237 47DA 5210 C690
C400 5742 2A03 02F0 5275
C408 A003 C5F1 F124 78C4
C410 1878 C4B0 79C3 0300
C418 10C4 3FD5 1164 1F38
C420 18D1 5A74 4E03 C4F1
C428 F124 10C4 4852 2410
C430 C43B D1D1 52D1 2C03
C438 D401 032C 125F 6111
C440 DF60 EFDF 1012 3400
C448 A71F C329 1037 0000
C450 FF1E F120 E421 D723
C458 CA25 BE27 B22A A72C
C460 9C2F 9232 8935 8138
C468 783B 713F 6A42 6446
C470 5D4A 574E 5153 4B58
C478 465F 4165 3C6C 3972
C480 3577 317E 2E85 0000
C488 FF36 08F7 04F8 00C0
C490 10F8 0A02 0000 311F
C498 117B D400 10C4 8800
C4A0 0788 1810 C690 8819
C4A8 F1E0 3700 0000 0000
C4B0 10C6 96D6 0728 0F57
C4B8 D2D2 D264 1F74 5403
C4C0 F8F1 F1D4 0037 0000
C4C8 6407 3402 80D1 D22F
C4D0 0237 0000 0000 0000
C540 0A0A 0908 0908 0700
C548 0607 0809 0A0A 0A00
C550 0A0A 0908 0908 0700
C558 0607 0809 0A0A 0A00
C560 10C6 E557 033B 8345
C568 DA11 DB52 F1E0 F1F9
C570 0200 0050 1F02 93C3
C578 9CF1 EA02 0634 FACF
C580 2467 0028 0302 11FA
C588 CFFF BC02 0434 2426
C590 2F03 2F15 F19A 10C6
C598 DC84 1B11 DB57 3410
C5A0 C6E5 5703 F8F1 E9F1
C5A8 F510 C6D3 5734 2426
C5B0 2F03 10C6 DC84 1A24
C5B8 10C6 DC84 1B10 F83B
C5C0 5202 5034 F1E0 2F03
C5C8 2F2A 10C6 E5D4 0037
```

```
1:"A"S=0:W=-500:C=&C56
  0: RESTORE 8
2:GOSUB 9: READ M: IF
  M=0 RESTORE 8: READ
  M
3:IF M=1 GOTO 49
4:POKE &C48F,M: CALL &
  C300:B= PEEK &C690:W
  =W+B: IF B=0 WAIT 0:
  A$="GO NEXT": CALL C
  : BEEP 0: BEEP 0:
  GOTO 2
5:WAIT 0:A=0: CALL C:A
  $="END": CALL C:
  BEEP 0: BEEP 0:A$="S
  CORE=": CALL C: BEEP
  0: PRINT A$;W: CALL
  4576: BEEP 0: BEEP 0
  : BEEP 0
6:IF H<W LET H=W
7:A$="HI SC=": CALL C:
  BEEP 0: PRINT A$;H:
  CALL 7437: GOTO "A"
8:DATA &C0,1, RND 48+7
  9,&C1,0
9:W=W+500:S=S+1:A$="ST
  AGE "+ STR$ S: WAIT
  0: PRINT " -","SCRAM
  BLE": WAIT 15: CALL
  C: BEEP 0: BEEP 0:
  RETURN
49:"=" CALL &BF18:B=5.1
51:CALL 4576: CALL &BF4
  0: GOTO 10B
52:W=W+ PEEK &8009:
  GOTO 5
53:A$="GO NEXT": WAIT 0
  : CALL C: BEEP 0:
  BEEP 0: GOTO 2
```

PC 1251

# CLASSEMENT ALPHABETIQUE

Ceci est un exemple de ce que peut apporter la rapidité du LM. En effet, cet utilitaire se charge de classer 200 mots en 27 secondes. On peut classer jusqu'à 255 mots de 12 lettres maximum.
Une fois le LM et le Basic entrés par les moyens habituels, faites RUN. Ensuite, entrez les mots à trier les uns à la suite des autres. Lorsque la liste est finie, tapez ENTER. Le classement s'effectue et s'achève quelques secondes après. Si vous voulez rajouter un mot : DEF S. Ecrivez-le puis ENTER ENTER. Il viendra prendre rapidement sa place dans la liste triée.

Le programme est facilement modifiable. Pour le transposer, modifiez, outre les adresses dans le Basic, les adresses des JP ET CALL en fin de programme (à partir de &B94E).

```
10:CLEAR : DIM B$(255)*
   12:N=-1
20:"S"N=N+1: INPUT B$(2
   55-N): IF N<255 THEN
   20
30:A=&C5C3-12N:B= INT (
   A/256):A=A-256B:N=N-
   1
40:POKE &B904,A: POKE &
   B908,B
50:CALL &B902: BEEP 1
55:USING "####"
60:"A" FOR I=0 TO N:
   PRINT I+1;"  ";B$(25
   5-N+I): NEXT I: END
```

```
TRI ALPHA
(C) G.NICOLAS

B8E0 0001 8413 1008 000B
B8E8 A004 1800 0184 1312
B8F0 0800 0BA0 0419 0001
B8F8 8413 1008 000B A004
B900 1937 9002 87DB 9102
B908 C5DB 9070 0C2A 0450
B910 7001 9163 C528 0751
B918 63C3 2802 3700 0113
B920 1092 0892 700C 2A04
B928 5070 0193 63C5 2806
B930 5163 CF39 2A00 0384
B938 1310 0888 020C DBA0
B940 0655 24C7 2804 4929
B948 0905 24C7 2A04 78B8
B950 E079 B923 0000 0000
```

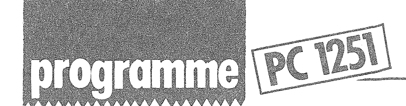


# BOMBER

Voici un petit programme pour PC-12xx. En fait il s'agit d'une tentative de performance dans la miniaturisation. Beaucoup de travail, pour 3 lignes de basic.
Le mode d'emploi est simple : Régler le contraste au maximum, faire RUN, attendre le BEEP (environ 25 secondes d'initialisation) et appuyez sur n'importe quelle touche (sauf ENTER, CL, des flèches, DEF et SHIFT) pour lâcher les bombes.

Philippe Pierson

```
10:G=0: FOR I=5 TO 59:C
   = RND 6:A=128*(1-2^-
   C): POKE &F800+I,A:
   NEXT I: BEEP 1: CALL
   &11E0: FOR H=1 TO 5:
   N=0:X=&F805
20:FOR I=4 TO 58: POKE
   X, PEEK X+1: IF
   INKEY$ <>"" LET G=G+
   7- LOG (129- PEEK X)
   / LOG 2: POKE X,1:N=
   N+1: IF N>4 LET I=58
30:POKE X, PEEK X-1:X=X
   +1: NEXT I: NEXT H:
   BEEP 2: PRINT "SCORE
   :",G: GOTO 10
```



# DESSIN ANIME

Voici un court programme LM pour PC 1251 de Johan Kovacs affichant un petit dessin animé.
Que représente-t-il ?
Vous ne le saurez qu'après l'avoir tapé sur votre poquette. La curiosité vaut bien un petit effort, non ?
Pour lancer le pgm : CALL & C 100
Pour arrêter le pgm : BRK

```
DESSIN ANIME
(C) J.KOVACS

C100 7811 E000 3B02 0010
C108 F800 1F00 3B11 401F
C110 1206 02FF DB50 02F7
C118 DB03 05C3 2804 79C1
C120 0078 C150 78C1 9078
C128 C180 78C1 6078 C190
C130 78C1 8078 C170 78C1
C138 9078 C180 6B08 3802
C140 3779 C11B 0000 0000
C148 0000 0000 0000 0000
C150 0244 2602 2226 021F
C158 2602 3426 0204 2637
C160 0224 2602 2226 021F
C168 2602 1226 0224 2637
C170 0214 2602 2226 021F
C178 2602 2426 0244 2637
C180 0707 0707 0702 0026
C188 2626 2626 3700 0000
C190 00C0 4128 0237 4EFF
C198 2D07 0000 0000 0000
```

# MANUEL LM 1251 (ERRATA)

**A la suite de quelques petites erreurs découvertes dans le « TECHNICAL MANUAL REFERENCE » du PC 1251, l'éditeur japonais de ce manuel nous a fait parvenir ce feuillet rectificatif que nous vous livrons « in extenso ».**

| PAGE | INCORRECT | CORRECT |
|---|---|---|
| 20 | ADDRESS / REGISTER<br>00 / I<br>≀ / ≀<br>04 / XH<br>05 / XL<br>06 / YH<br>07 / YL<br>≀ / ≀<br>09 / L | ADDRESS / REGISTER<br>00 / I<br>≀ / ≀<br>04 / XL<br>05 / XH<br>06 / YL<br>07 / YH<br>≀ / ≀<br>09 / L |
| 29 | ≀<br>320 : CALL & CA100<br>≀ | ≀<br>320 : CALL & C100<br>≀ |
| 31 | ≀<br>Addr Machine Code Label Mnemonic Operand(s) Comments<br>≀<br>*C10C 10 C6 A0 LIDP C6A1 "Window" address<br>≀<br>C112 78 C1 1D ① CALL convert high nibble<br>≀ | ≀<br>Addr Machine Code Label Mnemonic Operand(s) Comments<br>≀<br>*C10C 10 C6 A0 LIDP C6A0 "Window" address<br>≀<br>C112 78 C1 1D CALL ① convert high nibble<br>≀ |
| 33 | ≀<br>20 : INPUT "BYTES?" ; 3<br>30 : FOR I = 0 TO (3 – I)<br>≀ | ≀<br>20 : INPUT "BYTES?" ; B<br>30 : FOR I = 0 TO (B – I)<br>First enter and execute the program from line 400 listed on the next page, and then execute this program.<br>≀ |
| 34 | ≀<br>420: POKE &C10C, &10, &C6, &A0, &57, &34, &58, &78, &C1, &1C<br>430: POKE &C114, &5B, &78, &C1, &1C, &02, &00, &26, &37<br>440: POKE &C11C, &64, &0F, &34, &75, &0A, &3A, &06<br>450: POKE &C123, &5B, &74, &47, &2C, &04<br>460: POKE &C128, &5B, &74, &40, &26, &37 | ≀<br>420: POKE &C10C, &10, &C6, &A0, &57, &34, &58, &78, &C1, &1D<br>430: POKE &C115, &5B, &78, &C1, &1D, &02, &00, &26, &37<br>440: POKE &C11D, &64, &0F, &34, &75, &0A, &3A, &06<br>450: POKE &C124, &5B, &74, &47, &2C, &04<br>460: POKE &C129, &5B, &74, &40, &26, &37 |

# LE TABLEUR

Le mot tableur est peut-être un peu fort pour le programme que contiennent les PC-126x. Mais il permet au niveau d'un pocket la programmation de fonctions sans passer par une phase de programmation.
Pour utiliser le tableur, c'est très simple. On se met en mode PRO et on écrit sa fonction en la faisant précéder du caractère « # »
Pour bien utiliser ce programme nous allons vous donner quelques informations supplémentaires par rapport au manuel.
Tout d'abord, il est préférable de programmer des fonctions en minuscule. En effet, ainsi vous pourrez mettre des noms complets pour les variables sans risque qu'un mot clef soit reconnu.

**Exemple :**
# Maison = loyer + charges + gaz + eau + élec + téléphone.
# Remise = total *.2.
Pour le second exemple, s'il avait été saisi en majuscule, 1 mot clef aurait été reconnu : REM et TO. Une ligne de fonctions peut contenir jusqu'à 10 zones. Sur une ligne vous pouvez mettre 10 variables qui peuvent avoir de 1 à 7 caractères.
En exécutant manuellement un USING « # #... avant de lancer une fonction vous pourrez formater celle-ci.
Dans une fonction vous pouvez utiliser outre les opérations courantes (+, −, *, /, ^...) les instructions suivantes :

ABS ATN TAN EXP DEG SQR
INT ASN SIN LOG DMS PI
SGN ACS COS LN

Les instructions et opérateurs comme AND. OR. NOT. RND et PEEK... sont interdits.
**Voici quelques petites fonctions, faites nous parvenir les vôtres.**

```
#PMT= INT (.5+PV*i/(1-(1
 +i/100)^-n))/100:Total=
 PMT*n
#Vms=1000*Dist/(An*31557
 600+2629800*Mois+86400*
 Jour+3600*H+60*M+S):Vkh
 =3.6*Vms
#Difdate= INT (-(Jour1+3
 0.6*Mois1+365.25*Annee1
 )+(Jour2+30.6*Mois2+365
 .25*Annee2))
#Moy100k=Ljtre/Km*100
PMT: Calcul du paiement
      mensuel d un credit
Vms: Vitesse en fonction
      d une distance et
      d un temps
Difdate: difference en-
          tre 2 dates
Moy100k: moyenne kilome-
          trique
```



# START MOGURA

Vous allez en prendre plein la vue et les oreilles, heureux possesseurs de PC 1401, avec ce jeu en LM graphique et sonore. Basé sur les réflexes, il s'agit de dégommer les petites figures qui apparaissent au hasard sur l'une des huit premières matrices de l'écran. Pour ce faire, appuyer sur la touche de Z à SPC correspondant à la position sur l'écran.

**MODE D'EMPLOI :**

RUN. Répondez à la question SON (O/N) : par oui ou non (O ou N) selon votre option.
Réglez ensuite la vitesse (niveau de difficulté) : VITESSE 0-3 en accord avec votre dextérité. Puis, attendez le CHARGEMENT du LM et attention, ça part très vite.
Votre score s'inscrit ensuite à la fin du jeu et une pression sur ENTER permet de recommencer. On peut arrêter le programme par BRK.
Visez bien, pas d'affolement et surtout concentrez-vous.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Z | X | C | V | B | N | N | SPC |

```
  5:REM (C) I/O SEP 84
 10:POKE &2600,&1C,&24,&
    2C,&00,&07,&0D,&12,&
    16
 20:POKE &2720,&40,&40,&
    40,&40,&40,&70,&28,&
    08,&28,&70,&7E,&05,&
    01,&05,&7E
 30:POKE &272F,&44,&28,&
    00,&28,&44
 40:INPUT "SON (O/N):";A
    $:U=0:IF A$="O" LET
    U=&10
 50:INPUT "VITESSE 0-3:"
    ;V:V=&28+V*8
 60:PAUSE "     CHARGEME
    NT":CALL &05A2:S=&24
    FF
 70:READ M:IF M<999 LET
    S=S+1:POKE S,M:GOTO
    70
 80:PRINT "START    MOGU
    RA"
 90:CALL &2500:PRINT "PO
    INT=";PEEK &270B:
    GOTO 80
100:DATA &10,&27,&0C,&02
    ,&00,&52,&11,&0B,&52
    ,&11,&20,&00,&1F,&90
    ,&18,&E5,&A2,&10
110:DATA &27,&0C,&57,&42
    ,&52,&67,&32,&28,&02
    ,&37,&11,&00,&02,&10
    ,&00,&08,&1F,&78
120:DATA &25,&93,&78,&25
    ,&82,&4E,&FF,&43,&29
    ,&04,&78,&25,&82,&64
    ,&07,&10,&27,&0F
130:DATA &52,&03,&26,&E2
    ,&70,&24,&10,&27,&0E
    ,&52,&02,&15,&78,&25
    ,&4C,&02,&1A,&78
140:DATA &25,&4C,&2D,&3A
    ,&78,&25,&AD,&00,V,&
    E4,&8F,&8F,&DB,&10,&
    27,&0E,&57,&C7
150:DATA &38,&05,&41,&29
    ,&0D,&37,&5B,&5B,&02
    ,&1F,&78,&25,&AD,&10
    ,&27,&0B,&57,&42
160:DATA &52,&02,&00,&12
    ,&5F,&10,&25,&78,&61
    ,U,&DF,&4E,&00,&60,&
    01,&DF,&42,&52
170:DATA &29,&0B,&2D,&70
    ,&10,&27,&10,&84,&1B
    ,&85,&60,&9F,&61,&90
    ,&25,&10,&27,&10
180:DATA &84,&1B,&37,&02
    ,&00,&03,&60,&E2,&78
    ,&03,&27,&E2,&70,&03
    ,&05,&24,&30,&59
190:DATA &26,&50,&C3,&29
    ,&05,&84,&63,&08,&29
    ,&0E,&37,&34,&10,&27
    ,&0F,&57,&03,&27
200:DATA &E2,&78,&5B,&26
    ,&79,&25,&93,999
```

# PC 1401

**SUITE ET FIN DE NOTRE ARTICLE COMMENCÉ DANS LE N° 9**

# PC - L.M.

**• LE MICROPROCESSEUR**

Par bonheur, le microprocesseur qui équipe notre 1401 est le même que celui monté sur le PC 1251, beaucoup d'articles de nos bulletins y sont consacrés ; le manuel de référence du langage machine 1251 est églament applicable, pour le fonctionnement du microprocesseur, au PC 1401. La grande majorité des programmes L.M. du 1251 ne faisant pas appel à la ROM sont également adaptables en modifiant les adresses d'implantation en mémoire selon un principe très simple : bit 15 de l'adresse d'implantation = 0. Exemple : &C100 deviendra &4100.

**• QUELQUES PROGRAMMES**

**Liaison RUN-CALL**

```
65278 : « J » CALL &4100 : END
65279 : « K » CALL &4110 : END
&4100  00  07  10  46  A8  90  18  11
&4108  A0  90  19  37
&4110  00  07  10  46  A0  90  18  11
&4118  A8  90  19  37
```

DEF « J » fera passer la mémoire du mode « cal » dans la mémoire ANS du dernier résultat calculé en mode « run » (accessible par ↑ ou ↓ )

DEF « K » fera l'opération inverse

Ces programmes sont totalement relogeables.

**Inversion vidéo**

(demande des talents de cryptographe)

```
&4100  02  XX  03  YY  E2  70  24  F2  3E  67  FF  28  02  37  83  DB
&4110  59  00  00  67  08  3A  07  00  40  03  0F  45  59  82  44  44
&4120  83  44  DB  80  44  DB  03  60  E2  70  E2  78  02  04  34  24
&4130  83  60  00  42  45  DB  26  2F  09  F2  3E  2D  36
```

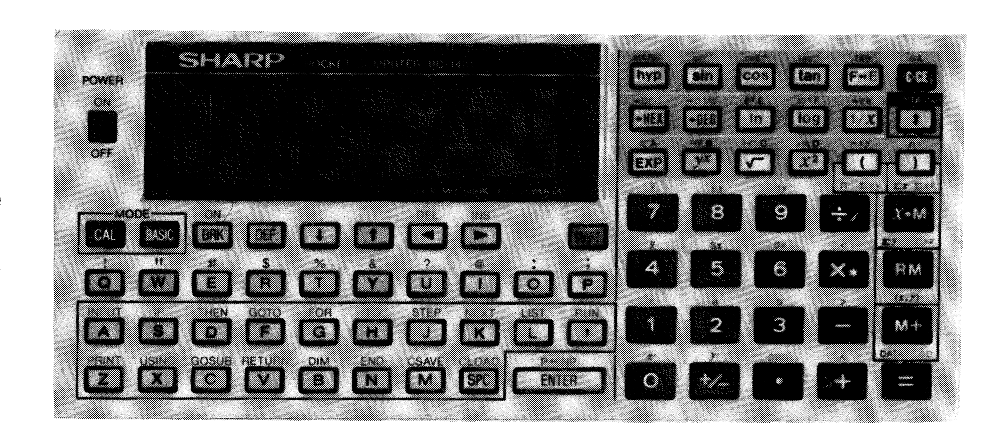
Mettez à l'adresse yyxx (soit « adresse YYXX ») et suivantes les numéros n des caractères à inverser (n entre 0 et 15 ; dimension de l'afficheur !) suivis de &FF (ne pas oublier).

Un CALL &4100 exécutera le programme (totalement relogeable). Si vous voulez voir quelque chose, n'oubliez pas de le faire précéder par un CALL &5A2 qui allume l'afficheur...

# PC CURIEUX

**Tripatouillage**

A titre de divertissement, essayez la séquence suivante (mode RUN) 123 HEX SIN SPEC ↓ (Miracle ! oh le joli dessin..) ; vous pouvez essayer de trouver d'autres variantes...






# PC PRATIQUE

**Coordonnées**

Le PC 1401 permet la transformation rectangulaire-polaire et réciproquement. La transformation cartésienne-cylindrique est immédiate ainsi que sa réciproque. Quant aux coordonnées sphériques, on a :

| | |
|---|---|
| (x,y,z) | (r,σ,ϱ) par R=POL(x,y):P=z:R=POL(R,H) |
| X,Y,H | R,Z,P |
| et | |
| (r, ,) | (x,y,z) par X=REC (R,T):H=Z:X=REC (X,P) |
| R,T,P | X,Z,H |

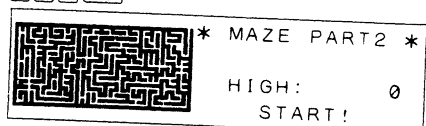



# LABY 2D

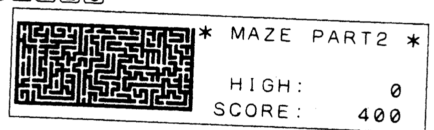
Revenu dans les temps les plus reculés de l'antiquité, vous voilà perdu au milieu d'un dédale digne d'Icare. En vous servant des touches 8 (pour monter), 2 (pour descendre), 4 (pour aller à gauche) et 6 (pour aller a droite) tentez de retrouver votre chemin.

SHARP
(extrait Tome 10)

❶ RUN ENTER

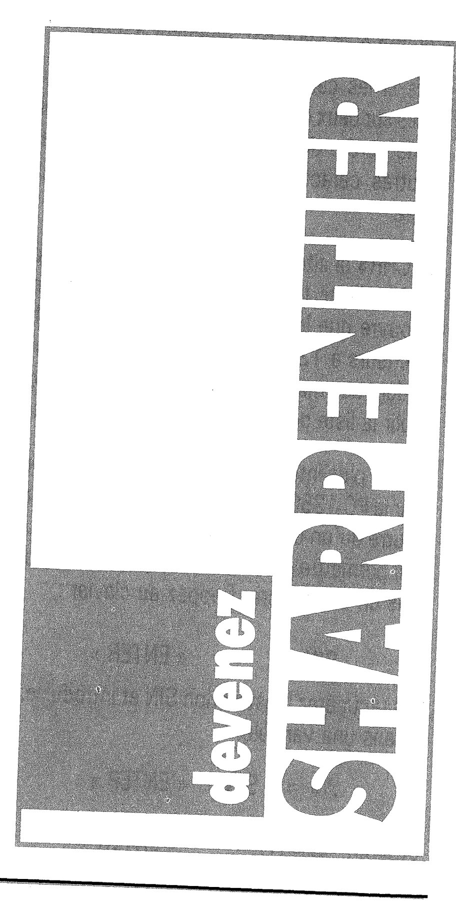



❷ 8 6 2 6




```
10:REM ++++++++++++++
20:REM + MAZE PART2 +
30:REM ++++++++++++++
40:CLEAR :DIM X(3),Y(3
   ),Z(32)
50:FOR I=0 TO 3: READ X
   (I),Y(I): NEXT I
60:DATA 0,1,-1,0,1,0,0,
   -1
70:"A" CLS : WAIT 0:
   CURSOR 10,0: PRINT "
   * MAZE PART2 *"
80:A=0:B=63:C=29:D=32:J
   =2: RANDOM
90:CURSOR 12,2: PRINT "
   HIGH:"; USING "#####
   #";H
100:CURSOR 10,3: PRINT "
    MAKING MAZE...":
    FOR I=0 TO 32:Z(I)=0
    : NEXT I
110:LINE (1,1)-(B,C),S,B
120:FOR I=3 TO 27 STEP 2
    : LINE (1,I)-(B,I),S
    ,&AAAA: NEXT I
130:FOR F=2 TO D-1: IF Z
    (F)=-1 LET A=A+1:Z(F
    )=A:I=F*2-1: PSET (I
    ,J): GOTO 190
140:IF J=C-1 AND Z(F)=0
    LET Z(F)=-1: GOTO 19
    0
150:IF RND 2>1 LET Z(F)=
    -1: GOTO 190
160:I=F*2-1: PSET (I,J):
    IF J<>C-1 GOTO 190
170:G=Z(F): FOR K=2 TO D
    -1: IF Z(K)=G LET Z(
    K)=0
180:NEXT K
190:NEXT F
200:J=J+1: FOR F=1 TO D-
    1: IF Z(F)=Z(F+1)
    AND Z(F)>-1 GOTO 290
210:IF RND 2>1 GOTO 290
220:I=F*2: PSET (I,J):
    IF Z(F)=Z(F+1) AND Z
    (F)=-1 LET A=A+1:Z(F
    )=A:Z(F+1)=A: GOTO 2
    90
230:IF Z(F)=-1 OR Z(F+1)
    =-1 IF Z(F)=-1 LET Z
    (F)=Z(F+1): GOTO 290
240:IF Z(F)=-1 OR Z(F+1)
    =-1 LET Z(F+1)=Z(F):
    GOTO 290
250:IF Z(F)>Z(F+1) LET L
    =Z(F):M=Z(F+1):
    GOTO 270
260:L=Z(F+1):M=Z(F)
270:FOR O=2 TO D-1: IF Z
    (O)=L LET Z(O)=M
280:NEXT O
290:NEXT F:J=J+1: IF J<C
    GOTO 130
300:X=2:Y=28: PRESET (62
    ,1):S=500:W=0
310:IF INKEY$ <>"" BEEP
    1: GOTO 350
320:IF W=0 CURSOR 10,3:
    PRINT "     START!
    ": PRESET (X,Y):
    BEEP 1
330:IF W=1 CURSOR 14,3:
    PRINT "          ":
    PSET (X,Y)
340:W=(W=0): GOTO 310
350:CURSOR 10,3: PRINT "
    SCORE:        "
360:PSET (X,Y),X
370:CURSOR 17,3: PRINT S
    :S=S-1: IF S=0 GOTO
    500
380:A= VAL INKEY$ : IF A
    <>0 AND INT (A/2)*2=
    A GOTO 400
390:PSET (X,Y),X: GOTO 3
    70
400:A=A/2-1
410:IF POINT (X+X(A),Y+Y
    (A)) GOTO 360
420:PRESET (X,Y)
430:X=X+X(A):Y=Y+Y(A)
440:IF Y>1 GOTO 360
450:BEEP 3
460:CLS : USING : PRINT
    "GOAL!"
470:PRINT "YOUR SCORE:";
    STR$ S
480:IF S>H PRINT "YOUR S
    CORE IS HIGHEST.":H=
    S
490:GOTO 510
500:BEEP 3: WAIT :
    CURSOR 10,3: PRINT "
    TIME OVER! ": CLS
510:WAIT 0: PRINT "AGAIN
    (Y/N)?"
520:A$= INKEY$ : IF A$="
    " GOTO 520
530:IF A$="Y" GOTO "A"
540:IF A$<>"N" GOTO 520
550:END
```

**1340bytes**



# UNE HISTOIRE DE CODES ASCII

Un code ASCII, personne ne me contredira, c'est une valeur de 0 à 255, donc sur un octet, qui caractérise un caractère. Frappons au clavier du PC 1350 : Z$=CRH$ 246, Z$=Z$+Z$ Z$=Z$+Z$

La variable Z$ devrait contenir 4 petits cœurs, et on n'en obtient que trois !

Non ce n'est pas une bogue, mais tout simplement le résultat d'une tronquature. En effet, tous les caractères dont le « code ASCII » est supérieur à 128 sont codés en réalité sur deux octets. Le premier code est FE (254 en décimal), le second est le code ASCII. Résultat, 4* 2 cela fait 8, et l'on dispose que de 7 octets, l'ordinateur à donc du tronquer la chaîne.

Donc pour les variables alphanumériques, il ne faut pas compter en caractères, la preuve, mais en octets :
1 octet si le code est inférieur à 128
2 octets si le code est supérieur à 128.

Mais si les caractères à partir de 128 sont codés sur deux octets, ce n'est pas pour le plaisir de perdre de la place, non, il doit exister d'autres caractères codés eux sur un seul octet.

Par CHR$ la situation est bloquée, il va donc falloir utiliser l'instruction POKE. Et là, on découvre que les caractères de codes ASCII supérieurs à 128, codés sur un octet correspondent aux mots clés du basic. On peut donc établir la liste des codes des fonctions basic.

Mais le plus intéressant, c'est finalement que pour le PC 1350, une fonction basic n'est rien d'autre qu'un caractères alphabétique, ce qui peut permettre pas mal de manipulations. Pour vous en convaincre, frappez au clavier :

SIN « ENTER »

puis récupérez l'instruction SIN et introduisez la dans une variable.

A$ = « SIN » « ENTER »

et enfin LEN A$ « ENTER »

Vous obtenez 1, votre variable ne contient que le caractère qui défini le mot clé, mais à l'affichage vous avez (SPACE) SIN (SPACE).

## CONCLUSION

Vous trouverez ci-dessous (ou dans les environs), la table des codes des mots clés du basic, ainsi que des tables de caractères obtenu en forçant le code FE ou au contraire en l'éliminant.

Je pense que l'on peut creuser encore dans cette voie, mais déjà ces résultats autorisent des manipulations intéressantes dont vous saurez tirer partie, j'en suis sûr.

Pascal Abrivard

```
                 !"#$%&'()*+,-./
0123456789:;<=>?@ABCDEFG
HIJKLMNOPQRSTUVWXYZ[¥]^_
`abcdefghijklmnopqrstuvw
xyz{|}¯ ----------------
¯LNLOGEXPSQRSINCOSTANINT
ABSSGNDEGDMSASNACSATNRND
ANDORNOTASCVALLENPEEKCHR
$STR$MID$LEFT$RIGHT$INKE
Y$PIMEMRUNNEWCONTPASSLIS
TLLISTCSAVECLOADMERGE¯¯O
PENCLOSESAVELOADCONSOLER
ANDOMDEGREERADIANGRADBEE
PWAITGOTOTRONTROFFCLEARU
SINGDIMCALLPOKECLSCURSOR
TOSTEPTHENONIFFORLETREME
NDNEXTSTOPREADDATAPAUSEP
RINTINPUTGOSUBAREADLPRIN
TRETURNRESTORECHAINGCURS
ORGPRINTLINEPOINTPSETPRE
SETBASICTEXTOPEN$¯0¥ABA♠
♥♦♣   π√¯
```

```
91 : LN
92 : LOG
93 : EXP
94 : SQR
95 : SIN
96 : COS
97 : TAN
98 : INT
99 : ABS
9A : SGN
9B : DEG
9C : DMS
9D : ASN
9E : ACS
9F : ATN
A0 : RND
A1 : AND
A2 : OR
A3 : NOT
A4 : ASC
A5 : VAL
A6 : LEN
A7 : PEEK
A8 : CHR$
A9 : STR$
AA : MID$
AB : LEFT$
AC : RIGHT$
AD : INKEY$
AE : PI
AF : MEM
B0 : RUN
B1 : NEW
B2 : CONT
B3 : PASS
B4 : LIST
B5 : LLIST
B6 : CSAVE
B7 : CLOAD
B8 : MERGE
B9 : ¯
BA : ¯
BB : OPEN
BC : CLOSE
BD : SAVE
BE : LOAD
BF : CONSOLE
C0 : RANDOM
C1 : DEGREE
C2 : RADIAN
C3 : GRAD
C4 : BEEP
C5 : WAIT
C6 : GOTO
C7 : TRON
C8 : TROFF
C9 : CLEAR
CA : USING
CB : DIM
CC : CALL
CD : POKE
CE : CLS
CF : CURSOR
D0 : TO
D1 : STEP
D2 : THEN
D3 : ON
D4 : IF
D5 : FOR
D6 : LET
D7 : REM
D8 : END
D9 : NEXT
DA : STOP
DB : READ
DC : DATA
DD : PAUSE
DE : PRINT
DF : INPUT
E0 : GOSUB
E1 : AREAD
E2 : LPRINT
E3 : RETURN
E4 : RESTORE
E5 : CHAIN
E6 : GCURSOR
E7 : GPRINT
E8 : LINE
E9 : POINT
EA : PSET
EB : PRESET
EC : BASIC
ED : TEXT
EE : OPEN$
EF : ¯
```

# L'INTERFACE SERIE DU PC 1350

Quelle interface série ? Et bien oui, celle de droite ou celle de gauche ? En effet le PC-1350 est équipé de 2 interfaces série. Une que nous rencontrons maintenant sur tous les PC (sauf 1500) et qui permet de communiquer avec les CE-124 et CE-126P. Elle se trouve sur le côté gauche. Sur le côté droit on peut rencontrer un connecteur 15 broches qui est aussi une interface série. Mais celles-ci à l'avantage d'être plus standard puisqu'il s'agit en fait d'une RS-232C.

## RS-232C...

Mais pourquoi ne pas avoir mis un connecteur normalisé ? Tout simplement parce que cette RS-232C n'entre pas dans les normes de l'EIA tel qu'elle est là. La RS-232C, c'est une interface série qui va comporter un certain nombre de paramètres qui répondent a une norme précise. Une des règles définissant cette norme Nous dit que les signaux, transmis sous forme d'une tension devront être compris entre −2 et −12 volts pour un « 0 » et entre +9 et +12 volts pour un « 1 ». Or le PC-1350 n'est alimenté qu'en 6 volts et les composants qui aurait pu augmenter cette tension lui aurait consommé trop de courant. C'est pourquoi Sharp a préféré laisser cette RS avec des niveaux de tension correspondant aux composants électroniques. C'est-à-dire 0 volt pour un « 0 » et 5 volts pour un « 1 » Ces niveaux sont TTL et c'est pourquoi il est dit de cette RS-232C du PC-1350 qu'elle est TTL. Alors ce petit connecteur 15 broche a été mis pour 2 raisons : tout d'abord parce qu'il est dangereux pour votre PC de le brancher sur une vraie RS-232C et puis aussi parce qu'un connecteur DB 25 c'est si gros que sur le PC-1350 cela n'aurait pas fait très joli.
Mais que peut-on faire avec une RS-232C TTL ? La connecter sur d'autres RS-232C TTL, car beaucoup de produits fonctionnent avec une telle RS mais là, c'est une autre histoire que nous verrons la prochaine fois.

# GESTION DE L'AFFICHEUR

## MELANGE DE TEXTE

Lorsque l'on affiche à l'écran à l'aide des fonctions PRINT, PAUSE ou INPUT, on efface une partie de ce qu'il y a l'écran.
Exemple :
10 LINE(0,0)-(149.31).X.BF :REM inverse l'écran
20 CURSOR 56 : PRINT « S » : efface l'écran à partir de la 56e position.
Ceci est du au fait que le PC travaille dans deux modes qui ne doivent normalement pas être mélangés : un mode texte et un mode graphique. pour des jeux, il est quand même intéressant de pouvoir afficher un score ou d'autres indications en graphiques. Plusieurs solutions s'offrent à vous :

- faire les graphiques en haut de l'afficheur de façon à pouvoir utiliser la ou les dernières lignes de l'écran. (cette méthode est utilisée dans le jeux que nous vous proposons).
- recréer en GRINT dans un tableau les caractères que vous voulez afficher.
- utiliser le sous-programme suivant qui affichera la chaine A$ à la position X.

A bientôt pour d'autres utilitaires graphiques.

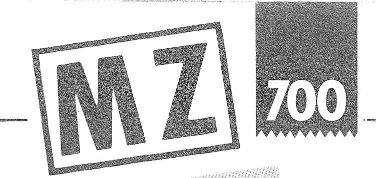
```
5:Y= INT (X/24)*8:X=(X
  -Y*3)*6+6: GCURSOR (
  X,Y+7): FOR I=1 TO
  LEN A$:S= ASC MID$ (
  A$,I,1)*5+&7FEE
6:WAIT 0: FOR J=S TO S
  -4 STEP -1: GPRINT
  PEEK J;: NEXT J:
  GPRINT 0;: NEXT I:
  RETURN
```

```
5:Y= INT (X/24):X=(X-Y
  *24)*6+6:Y=(Y+1)*8-1
  : GCURSOR (X,Y):
  FOR I=1 TO LEN A$:S=
  ASC MID$ (A$,I,1)*5+
  &7FEE
6:WAIT 0: FOR J=S TO S
  -4 STEP -1: GPRINT
  PEEK J;: NEXT J:
  GPRINT 0;: NEXT I:
  RETURN
```

## INVERSEZ DES MOTS

Dans certaines applications, il est parfois intéressant d'attirer l'attention de l'œil sur une partie d'un texte. La solution la plus simple est de passer un coup de surligneur (pour ne pas citer une marque). Sur un écran on pourra surligner en passant le mot désiré en inversion. Le sous-programme permettant ceci est en ligne 290, le reste n'étant là que pour vous montrer ce que l'on peut faire avec : vous pouvez déplacer le curseur (zone inversée) ou vous voulez sur l'afficheur grâce aux touches 8, 2, 4 et 6.

```
200:"=" WAIT 0: CLS :
    PRINT "ABCDEFGHIJKLM
    NOPQRSTUVWXYZ.012345
    6789()/:*;-,^<>@√π?&
    %$#! ="; 
210:PRINT "ABCDEFGHIJKLM
    NOPQRSTUVWXYZ.012345
    6789"
220:X=0:Y=0:L=4: GOSUB "
    cur"
230:A= VAL INKEY$ : IF A
    =0 OR A/2<> INT (A/2
    ) THEN 230
240:IF A=2 AND Y+1<4
    GOSUB "cur":Y=Y+1:
    GOSUB "cur": GOTO 23
    0
250:IF A=8 AND Y-1>=0
    GOSUB "cur":Y=Y-1:
    GOSUB "cur": GOTO 23
    0
260:IF A=4 AND X-L>=0
    GOSUB "cur":X=X-L:
    GOSUB "cur": GOTO 23
    0
270:IF A=6 AND X+L<=24-L
    GOSUB "cur":X=X+L:
    GOSUB "cur": GOTO 23
    0
280:BEEP 1: GOTO 230
290:"cur"V=5+X*6:W=Y*8:
    LINE (V,W)-(L*6+V,W+
    7),X,BF: RETURN
```




# LANGAGES/FORTH

**FORTH, véritable langage du « quatrième type », comme l'indique l'origine anglo-saxonne de son nom (FOURTH), est apparu à la fin des années 60.**

**Son créateur CHARLES HENRY MOORE définit sa principale possibilité comme étant d'« ajouter n'importe quelle fonction lorsqu'elle se révèle nécessaire ».**

En effet, toute la magie de FORTH réside dans cette faculté de rajouter des fonctions, des mots FORTH, lorsqu'on en a besoin. Le langage FORTH peut donc s'assimiler à une boîte à outils qui permet à son utilisateur de créer de nouveaux outils au fur et à mesure de ses besoins. Et encore mieux, ces nouveaux mots ou outils FORTH pourront, eux aussi, servir à en construire d'autres, et ainsi de suite...

En résumé, on peut dire qu'un mot FORTH est bâti sur le principe des « matrioschkas », vous savez, ces petites poupées russes qui s'emboitent les une dans les autres.

FORTH est donc un langage sans programme, les mots créés pour réaliser une application précise remplacent tout à fait avantageusement celui-ci.

Avant de traduire en exemples concrets tout ce que nous venons de dire, continuons d'explorer les principales caractéristiques du langage FORTH.

• FORTH met sur une « pile » toutes les données qu'il manipule; ce que fait également votre BASIC préféré, mais sans vous le dire... Cette pile fonctionne exactement comme une pile d'assiettes sur laquelle vous déposez la vaisselle fraîchement lavée, c'est-à-dire vos données. Vous les reprenez ensuite, et la dernière rangée sur le dessus de la pile se trouve alors la première reprise.

• FORTH range dans son « dictionnaire » tous les mots que vous pouvez être ammené à créer lors de vos applications. Un FORTH, telle la version d'EDIFORTH qui vous est présentée ci dessous, comprend environ 300 mots dans son dictionnaire de base.

• Aussi bien pour lui même que pour les applications qui sont construites avec, FORTH est très peu gourmand en mémoire; ainsi, EDIFORTH et ses 300 mots n'occupe que 10,5 Koctets un MZ 700.

• Parlons maintenant rapidité d'exécution : EDIFORTH, sur un MZ 80K parcourt une boucle de 1 à 10 000 en 1 seconde; en BASIC 5025, 14 secondes sont necessaires pour effectuer le même travail. Enfin, FORTH offre la possibilité d'écrire des jeux vidéo avec une rapidité d'affichage proche du langage machine et la facilité d'utilisation d'un langage évolué.

• La façon dont FORTH gère la mémoire de l'ordinateur et les mémoires extérieures de ce dernier est particulièrement remarquable. En effet, pour FORTH, la ou les mémoires qu'il utilise constitue une suite d'écrans qui peuvent être affichés et modifiés à volonté et sont gérés sous forme de mémoire virtuelle; c'est-à-dire que lorsque vous désirez disposer d'un écran précis, FORTH saura toujours où le retrouver, soit dans la mémoire interne de l'ordinateur, soit dans sa mémoire de masse extérieure : disquette ou cassette, par exemple.

## COMMENT FONCTIONNE FORTH.

La figure (1a) représente l'introduction des nombres 10 et 4 sur la pile ("" pour le retour chariot "CR"). L'interpréteur FORTH renvoie le message de bonne fin "(OK)" si aucune erreur n'a été détectée. Le symbole ">" est un marqueur de début de ligne qui permet à l'éditeur d'EDIFORTH de prendre en compte des lignes de mots pouvant avoir une longueur maximale égale à un écran. Contrairement à EDIFORTH, le FIG-FORTH standard ne connait que des lignes de 64 caractères, ignore les touches de déplacement du curseur ainsi que le mode insertion et les touches « AUTO-REPEAT ».

La figure (1b) vous montre l'évolution de la pile après chaque introduction. Comme vous pouvez le constater, après le retour chariot, la pile est vidée automatiquement, là aussi, il s'agit d'une spécificité d'EDIFORTH. Rassurez vous, ce langage est stylé, il n'effectue le « ménage » qu'après le départ du dernier invité, comme vous le montrent les figures (2a) et (2b).

En effet, nous venons de faire connaissance avec notre premier invité FORTH, pardon, MOT FORTH, en la personne de ".". Ce mot a pour fonction d'afficher le nombre situé au sommet de la pile. Celui-ci est effacé de la pile après son affichage à l'écran. Le nombre 10 qui n'est pas affiché par un mot "." est, lui aussi, effacé de la pile lors de la fin de l'interprétation et du passage en mode attente commande.

Maintenant que nous savons comment fonctionne cette pile, nous pouvons étudier le mot "*", bien connu des habitués du BASIC. Comme vous le montrent les figures (3a) et (3b), ce mot multiplie entre eux les 2 nombres du sommet de la pile, les efface et dépose le résultat sur cette dernière.

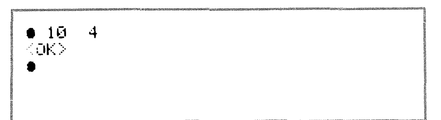



Figure 1a

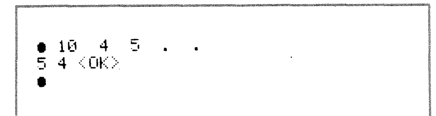



Figure 2a

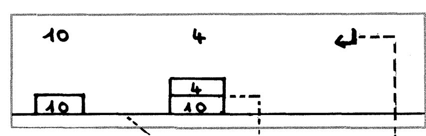



Figure 1b FOND DE LA PILE LA PILE RETOUR CHARIOT

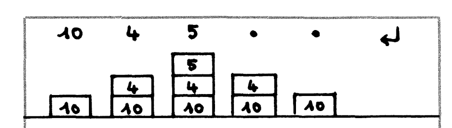



Figure 2b

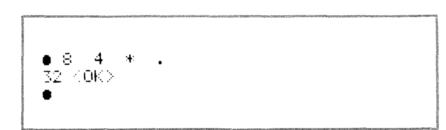
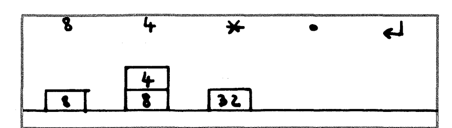



Figure 3a


Figure 3b

```
● 10 SURLAPILE
Le nombre 10 est au sommet de la pile
 OK>
●
```
Figure 6b

```
( SOUND          ecran no 68 ... 02/11/84 )
: SOUND    0 DO 2DUP MSTA MSTA
              LOOP DROP DROP MSTP ;

( < n1,n2,n3 --- > genere 'n1' fois un
  son compose deux sons primaires dont
  les tonalites sont reglees par 'n2' et
  'n3' )
: SOUND2   0 DO 2DUP MSTA MSTA 2DUP 2 /
                MSTA 2 / MSTA 2DUP MSTA
                MSTA
             LOOP DROP DROP MSTP ;
: SOUND3   0 DO 2DUP MSTA MSTA SWAP 2DUP
                2 / MSTA 2 / MSTA SWAP
                2DUP MSTA MSTA
             LOOP DROP DROP MSTP ;
( < n1,n2,n3 --- > fonctionnent comme
  'SOUND' )
```
Figure 9

Vous préférez les textes aux nombres ? Qu'à cela ne tienne, le mot "." vous le permet ainsi que vous pouvez le remarquer sur la figure (4).

Arrivé à ce stade, une précision s'impose : Un mot FORTH doit toujours être séparé de ce qui le suit par, au moins, un espace. Ce dernier est le seul séparateur reconnu par ce langage; ce qui vous explique la présence d'un espace entre le mot, et le message « BONJOUR ». Comme pour le mot « PRINT » du BASIC, le texte à afficher doit se terminer par « " ».

A propos de « PRINT », pourquoi ne pas remplacer le mot « . » pas très significatif, avouons-le, par « PRINT » ou « AFFICHE » ? Rien de plus simple, à l'aide des nouveaux mots FORTH « : » et « ; » qui indiquent le début et la fin de la définition d'un mot. Ainsi que l'indique la figure (5), il est facile, en FORTH de faire effectuer la même chose à un nouveau mot, que celle faite par un mot déjà existant dans le dictionnaire. L'avantage est que ce mot est créé par vous avec le nom que vous souhaitez; neammoins, le mot d'origine existe encore et peut toujours être utilisé.

Comment, arrivé à ce stade de la découverte de FORTH, résister à l'envie de créer de nouveaux mots ? C'est pourquoi nous vous livrons, sans commentaires, les figures (6a) et (6b).

```
● ." BONJOUR "
BONJOUR <OK>
●
```
Figure 4

```
● : PRINT  .  ;
<OK>
```
Figure 5

```
● : SURLAPILE  ." Le nombre " PRINT
    ." est au sommet de la pile   " ;
<OK>
●
```
Figure 6a

Maintenant, pour ceux qui veulent savoir ce qu'ils peuvent faire avec FORTH sur leur MZ, voici 3 exemples d'écrans extraits de la disquette MASTER EDIFORT en version MZ 80K.
- Ecran N4 (figure 7) :
Celui-ci vous présente un exemple d'utilisation de la récursivité en FORTH.
Tout d'abord, il faut créer un mot « MYSELF » qui permette à un mot de s'appeler lui-même. Ensuite, pour tester le mot « MYSELF », créons un mot « FACT ». Celui-ci calcule, en s'appelant lui-même, la factorielle du nombre « n » et dépose cette dernière sur le sommet de la pile.

```
( MYSELF,FACT  ecran no 4 ... 03/11/84 )
: MYSELF   LATEST PFA CFA , ; IMMEDIATE

( < --- > autorise la recursivite en
  permettant , dans la definition d'un
  mot que ce dernier puisse  s'appeller
  lui-meme )

: FACT    -DUP
          IF DUP 1- MYSELF *
          ELSE 1
          THEN ;

( < n --- n! > calcule la factorielle
  'n!' de 'n' par recursivite a l'aide
  du mot 'MYSELF' )
```
Figure 7

```
( MSTA , MSTP ecran no 67 ... 02/11/84 )
BASE @ HEX
CREATE MSTA   E1 C, 22 C, 11A1 , CD C,
              0044 , C3 C, 1244 ,
              0A ALLOT   SMUDGE

( < n --- > mot generant un son
  ininterrompu dont la tonalite est
  fonction de la valeur 'n' )

: MSTP   0047 CALL ;

( < --- > mot arretant un son genere par
  le mot 'MSTA' )

BASE !
```
Figure 8

Les écrans n° 67 et 68 (Fig.8 et 9) vous indiquent comment un ensemble de mots permet de « préfabriquer » des bruits de lasers et autres explosions, avec votre MZ.
Ainsi, en « préfabriquant » un certain nombre de fonctions souvent utilisées dans les jeux vidéo, vous pourrez, par exemple, arriver à réaliser ce type d'application très rapidement. Et, n'oubliez pas... UN MOT FORTH, UNE FOIS CREE, EST TOUJOURS REUTILISABLE ULTERIEUREMENT...
Il serait sans doute souhaitable de décrire le fonctionnement de chacun des mots FORTH de base utilisés pour créer les mots des 3 écrans précédents. Alors, si vous désirez, vous aussi, devenir FORTH en thème, écrivez-nous vite au club, une série d'articles d'initiation au langage FORTH sera alors envisagée.

Un dernier mot pour vous signaler que, dès février 85, le Club sera en mesure de diffuser, pour votre MZ :
• EDIFORTH- 1.2 version cassette pour MZ700 au prix de 100 F
• EDIFORTH- 2.2 version cassette pour MZ80K au prix de 100F.
• EDIFORTH- 2.6 version disquette pour MZ80K au prix de 150F.
• EDIFORTH- 3.2 version cassette pour MZ80A au prix de 100F.
Leurs principales caractéristiques sont les suivantes :
• Gestion de la couleur et du 2e jeu de caractères (MZ700)
• Gestion complète de la table traçante (MZ700) ou imprimante (80)
• Editeur plein écran et clavier auto-repeat. (Nettement supérieur au FORTH standard)
• Mots de gestion du son et de l'horloge interne.
• Gestion des poignées de jeux (700).
• Pratiquement compatible à 100% avec le FORTH standard (FIG. FORTH
• Dictionnaire de 300 mots.

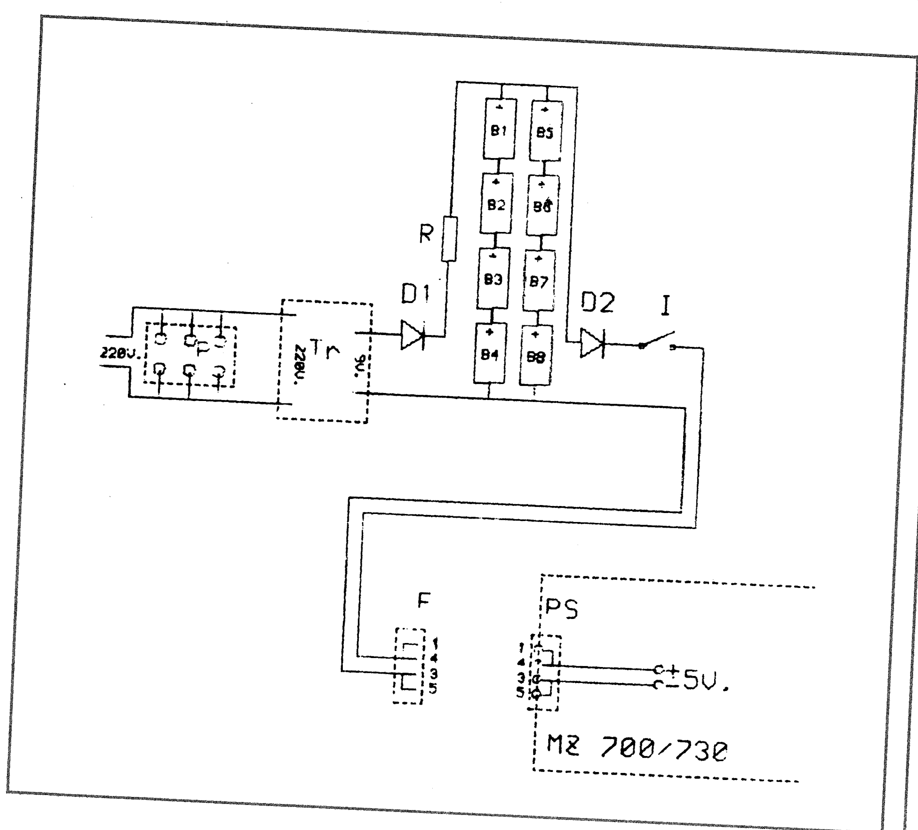
Dominique BEURRIER

# ALIMENTATION DE SAUVEGARDE

C'est bien connu, la plus grande faiblesse d'un ordinateur de table, c'est sa prise de courant. Il suffit que ce cher 220 volts nous manque pendant une toute petite fraction de seconde pour que plusieurs heures de travail soient réduites à néant et que notre petit cœur soit très ébranlé par l'inévitable crise de nerfs qui suit ce que les spécialistes appellent une micro-coupure. Heureusement, les Sharpentiers sont là, nous vous présentons, en effet, le shéma et les explications concernant une alimentation de sauvegarde qui prendra automatiquement le relai des éventuelles faiblesses de votre secteur.

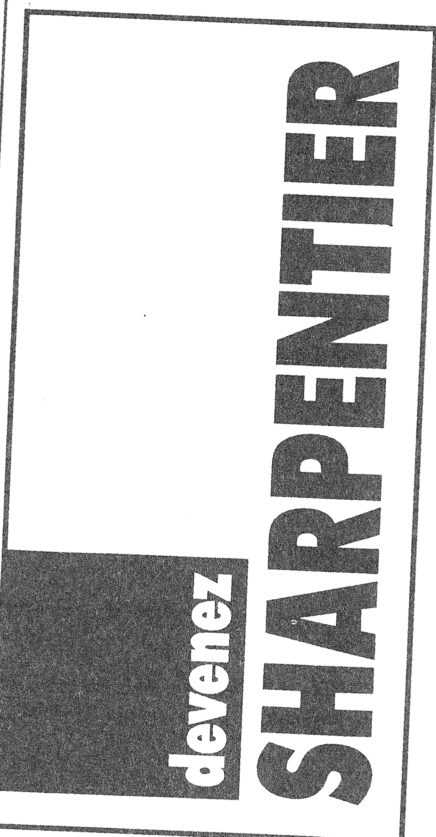
## UTILISATION :

Brancher, dans l'ordre, l'alimentation au réseau, interrupteur ouvert, l'ordinateur et le moniteur sur la prise multiple ; ensuite, la fiche F sur la prise PS. Mettre votre MZ en marche puis fermer l'interrupteur de l'alimentation. En fin d'utilisation, procéder en sens inverse. Cette alimentation de secours fonctionne depuis 3 mois chez son concepteur MARC LUCAS, permet une sauvegarde de 28 minutes maximum, et vous coûtera environ 300 F. Elle est, bien sûr, adaptable sur tout MZ.




P : Prise secteur multiple.
TR : Transformateur 220V/9V 1 Ampère.
D1 : Diode 1 ampère.
D2 : Diode 5 ampères.
R : Résistance 33 Ohms/1watt
I : Interrupteur 3 ampères.
F : Fiche mâle DIN 5 broches.
PS : Prise socle DIN 5 contacts.
B1 à B8 : Accumulateurs CADMIUM NICKEL 1,2 V/0,45 Ah type NC R6 de SAFT.
Coffret au choix.

Les fils reliant la prise PS au + et − de l'ordinateur sont à brancher sur le connecteur 2 broches situé en bout des cables sortants de l'alimentation. Respecter les polarités, c'est une question de vie ou de mort pour votre MZ. Avant une première mise en service ou après une inutilisation de plusieurs mois ou après un fonctionnement en sauvegarde de plus de 15 minutes, brancher cette alimentation sur le secteur, INTERRUPTEUR OUVERT, pendant 15 heures environ.






# ORDRE COMPACT POUR LE K-BASIC V.5.2.

Nous allons, cette fois-ci, ajouter au K-Basic V.5.2. un instruction du Super-Basic, l'ordre COMPACT. Son utilité est la suivante :
Supposons que nous écrivions un programme sous forme structurée avec identation des lignes (pour mieux en faire ressortir la structure) ét séparation des mots-clés par des espaces pour le rendre plus lisible. Lorsque le programme est parfaitement au point, il peut être devenu trop gourmand en mémoire au point de ne plus permettre l'utilisation de nombreuses variables ou de grands tableaux. On pourra alors utiliser COMPACT pour obtenir une version « utile » du programme, ne comportant plus d'espaces entre les mots clés et dans laquelle, toutes les indentations ont été supprimées. Cependant, les lignes contenant des REM sont conservées, pour que d'éventuels commentaires utiles ne soient pas perdus (on remarquera que ces commentaires sont compactés eux aussi (!) à moins qu'on les écrive entre "").

Pour ajouter cet ordre au K-Basic V.5.2., on le chargera et par Shift Break, on retournera au moniteur en mémoire morte. Grâce à la commande M, on placera dans les octets

| n° | la valeur | |
|---|---|---|
| IOFD | 33 | (Identificateur : "K-Basic V.5.3") |
| 83F9 | FD | (6FFD est l'adresse de la routine) |
| 83FA | 6F | |
| 7F35 | 43 | ASCII de C |
| 7F36 | 4F | ASCII de O (COMPACT aura le |
| 7F37 | 4D | ASCII de M code FE AD) |
| 7F38 | 50 | ASCII de P |
| 7F39 | 41 | ASCII de A |
| 7F3A | 43 | ASCII de C |
| 7F3B | D4 | ASCII de T + 80H |
| 7F3C | 42 | ASCII de B (On doit décaler |
| 7F3D | 4F | ASCII de O l'ordre suivant |
| 7F3E | 4F | ASCII de O qui est BOOT) |
| 7F3F | D4 | ASCII de T + 80H |
| 7F40 | FF | Fin de table des codes FE... |

On a donc diminué légèrement la zone FREE2 qui commence maintenant à 6D41 (On remarquera que 6D41 + 1200 = 7F41). Il reste à écrire la routine grâce à M à partir de 6FFD+1200=81FD.

On entre les codes suivants à partir de 8IFD :

```
CD 64 2C 2A A1 73 54 5D D5 7E ED AO
B6 ED AO 20 08 ED 53 A3 73 E1 C3 47
18 ED AO ED AO 7E FE 3A 28 04 FE 20
20 09 23 18 F4 23 7E FE 20 28 FA B7
28 2A FE 22 20 0C ED AO 7E B7 28 20
FE 22 28 18 18 F4 FE OB 28 OE FE II
28 OA FE 15 20 OA ED AO ED AO ED AO
ED AO ED AO ED AO 18 CE ED AO 4D 44
EB DI E5 ED 52 ED 7A B7 20 OB 7B FE
05 20 06 EI ED 52 EB 18 04 73 23 72
DI 69 60 19 8B
```

Avant de préenregistrer le Basic par J8670 et enfoncement de 'S', on revérifiera soigneusement les codes entrés (le dernier doit être entré dans l'octet n° 8279).

KOKANOSKY

# TRAITEMENT DE TEXTES

**Ce traitement de textes, écrit pour MZ 700 utilisera au mieux vos nouveaux caractères accentués. On peut le modifier assez facilement pour le faire tourner sur MZ 80K ou A.**

## A. MODES ET FONCTIONS

1. Définition du format d'écriture
2. Fixation des tabulateurs
3. Définition de l'autoreverse
4. Création des mots-type (F1 à F10)
5. Ecriture et édition à l'écran
6. Edition et restructuration
7. Impression du texte
8. Sauvegarde sur cassette
9. Lecture d'une cassette
10. Retour au Basic

## B. UTILISATION

**1. Format**

Entrée du nombre de caractères de l'imprimante, de la marge, du nombre de caractères du texte, du nombre de lignes par page, du retour automatique à la ligne.

**2. Tabulateurs**

Possibilité de 10 tabulateurs simultanés, à gauche ou à droite. Par défaut la valeur des tabulateurs est fixée à 0

**3. Autoreverse**

Cette fonction permet de provoquer un retour à la ligne dès l'apparition d'un espace ou d'un trait d'union dans les 8 derniers caractères précédant la fin de la ligne.
Par défaut R est pris égal à 3.

**4. Mots type**

Il est possible de programmer 10 mots ou expressions utilisés fréquemment sur les touches de fonction F1 à F10, ces mots-types ont 15 caractères maximum.

**5. Ecriture et édition**

Ce mode permet l'écriture du texte à l 'écran, il utilise les symboles ci-après :
£ : retour au menu
⟶ : Tabulateur gauche




←: Tabulateur droite
↓ : centrage
— : Correction (touche sans marquage)
\ : Répétition

Ces fonctions permettent une édition avec correction simple.

**6. Edition/Restructuration**

Ce mode permet la correction de lignes déjà complètes, l'effacement de lignes et leur déplacement.

Une possibilité de coupure automatique des lignes est prévue.

**7. Impression**

Assure l'impression de tout ou partie d'une page de texte.

**8. Sauvegarde sur cassette**

Permet de conserver une page de texte sur cassette.

**9. Lecture cassette**

Permet la lecture d'une cassette contenant une page de texte, en vue de sa réutilisation : impression, modification etc.

**10. Sortie programme**

Permet le retour au Basic.

Attention : dans ce cas toutes les informations (texte, format, mots-type) sont perdues !

## C. CONCLUSION

Ce programme n'a pas la prétention à une qualité professionnelle, et à un certain nombre d'imperfections.

Il permet néanmoins de rendre de grands services à l'amateur.

La capacité mémoire permet le stockage de 10 pages (70 lignes, 80 caractères).

Options par défaut :

- Imprimante et texte : 40 colonnes
- Marge gauche : 0 autoreverse : 3
- Tabulateurs gauche et droite : 0 nombre de lignes par page : 70.

H. Benoit

```
0 REM*********traitement de texte********
20 FOR I=4653 TO4676 :REM CLAVIER MACHINE A ECRIRE QWERTY
30 P=PEEK(I):O=PEEK(I+64)
40 POKEI,O:POKEI+64,P
50 NEXTI
60 DATA4645,189,4709,89,4646,162,4710,90,4686,185,4750,168,4711,173,4712,174,471
3,107,4648,178,4647,187,4749,110,4649,186,4753,109,4743,111,4744,106,4745,105
70 FOR I=1 TO 17
80 READ P,W:POKE P,W:NEXT I
85 ZZ=40:IP=40:LR=0:AR=37:ZS=70:RL$="n"
90 PRINT"■"
100 DIM S$(70,20):DIM LI(10):DIM RE(10):IF ZS<2:ZS=70:REM DEF. DE BASE
110 IFZZ<2:ZZ=40
120 IFLR<0:LR=0
130 IFAR<2:AR=41
140 PRINT"*****TRAITEMENT DE TEXTE MZ-700*******":PRINT
150 PRINT"******COPYRIGHT BY: ROLF KAPPLER*******"
155 PRINT"*****Publie par HOMECOMPUTER 10/84*****"
160 PRINT"****Traduit et adapte par H.BENOIT*****":PRINT
190 PRINT"         Octets disponibles:";SIZE:PRINT
200 PRINT"********CHOIX DU SOUS-PROGRAMME********":PRINT
210 PRINT"1:FORMAT D'ECRITURE"
220 PRINT"2:TABULATEUR"
230 PRINT"3:AUTOREVERSE"
240 PRINT"4:MOTS-TYPES"
250 PRINT"5:COMPOSITION ET EDITION"
260 PRINT"6:EDITION/RESTRUCTURATION"
270 PRINT"7:IMPRESSION"
280 PRINT"8:SAUVEGARDE SUR CASSETTE"
290 PRINT"9:LECTURE CASSETTE"
300 PRINT"10:SORTIE PROGRAMME":PRINT
310 ON ERROR GOTO 3010
320 INPUT"CHOISIR UN NOMBRE S.V.P. ";MW
330 ON ERROR GOTO 0
340 ON MW GOTO350,440,840,2750,910,1480,2590,2280,2380,2710
350 PRINT"■":PRINT"          ***FORMAT D'ECRITURE***":PRINT:ZS=70
360 PRINT"ENTRER LES NOMBRES CORRESPONDANT         A VOTRE CHOIX S.V.P.    ":PRINT
370 ON ERROR GOTO 3010
375 INPUT"CARACTERES/LIGNE (IMPRIMANTE): ";IP
380 INPUT "ESPACES POUR LA MARGE GAUCHE: ";LR
390 INPUT"CARACTERES/LIGNE (HORS MARGE): ";ZZ
395 IF LR+ZZ>IP:PRINT"MAXIMUM";IP;" CARACT. PAR LIGNE!":GOTO390
400 INPUT"LIGNES PAR PAGE:";N:GOSUB3040:ZS=N
410 ON ERROR GOTO 0
415 INPUT"RETOUR LIGNE AUTOMATIQUE? (o/n) ";RL$
430 GOTO90
440 RE(0)=LR+ZZ:I=1:REM TAB.
450 PRINT"■":PRINT"              ***TABULATEUR***":PRINT
470 INPUT"   GAUCHE(g)ou DROIT(d)? ";X$
480 IF X$="g" GOTO 510
490 IF X$="d":GOTO 630
500 GOTO 90
510 PRINT:PRINT"10 POSITIONS DE TABULATEUR POSSIBLES■"
515 PRINT"DONNER LEUR NOMBRE D'ESPACES CROISSANT■"
520 PRINT"APRES LA DERNIERE ENTREE PRESSER %0'■"
530 PRINT"PROGRAMME QUITTE AU";L;"EME CARACTERE■"
540 ONERRORGOTO3010
550 PRINTI;"E. POSITION: ANCIENNE VALEUR";LI(I)
560 INPUT LI(I)
570 ON ERROR GOTO 0
580 IFLI(I)=0:FORI=ITO10:LI(I)=0:NEXTI:GOTO90
590 IFLI(I)<=LI(I-1):PRINT"ENTREE ILLOGIQUE":GOTO540
600 IF ZZ<=LI(I):PRINT"ENTREE ILLOGIQUE":GOTO 540
610 I=I+1:IFI<=10:GOTO540
620 GOTO 90
630 PRINT:PRINT"10 POSITIONS DE TABULATEUR POSSIBLES■"
635 PRINT"DONNER LE NOMBRE D'ESPACES DECROISSANT■"
640 PRINT"APRES LA DERNIERE ENTREE PRESSER %0'■"
650 PRINT"PROGRAMME QUITTE AU";L;"EME CARACTERE■"
660 ONERRORGOTO3010
670 PRINTI;"E. POSITION: ANCIENNE VALEUR";LI(I)
680 INPUT RE(I)
690 ON ERROR GOTO 0
700 IFRE(I)=0:FORI=ITO10:RE(I)=0:NEXTI:GOTO90
710 IFRE(I)>=RE(I-1):PRINT"ENTREE ILLOGIQUE":GOTO660
720 IF RE(I)>=ZZ:PRINT"ENTREE ILLOGIQUE":GOTO 660
730 I=I+1:IFI<=10:GOTO660
740 GOTO 90
750 GOSUB 1290 :REM CENTRAGE
760 GOSUB 3110
770 M$=S$(J,ST)+M$
780 AM=ZZ/2-(LEN(M$)-LR)/2
790 I=0
800 IF I>=AM:GOTO830
810 M$=" "+M$
820 I=I+1:GOTO 800
830 S$(J,ST)=M$:L=LEN(S$(J,ST))-1:GOTO1190
840 PRINT"■              ***AUTOREVERSE***":PRINT
850 PRINT"NOMBRE DE CARACTERES A INTERCALER PAR L'AUTOREVERSE A PARTIR DE LA MAR
GE DROITE"
860 ON ERROR GOTO 3010
870 INPUT R
880 ON ERROR GOTO 0
890 AR=LR+ZZ-R
900 GOTO90
910 PRINT"■":REM UP. 2
920 ON ERROR GOTO 3010
930 INPUT"QUELLE PAGE?";ST:PRINT"■"
940 GOSUB 3070
950 ON ERROR GOTO 0
960 FOR J=1 TO ZS          :REM LIGNES
970 R=0:LT=0:RT=0:PRINT
980 IF S$(J,ST)="":GOTO 1020
985 GOTO1000
990 IF LEN(S$(J,ST))>39:PRINTS$(J,ST);"■■":GOTO1010
1000 PRINT S$(J,ST);"_■■"
1010 L=LEN(S$(J,ST)):GOTO 1030
1020 GOSUB 1290:PRINTTAB(LR);"_■■":L=LR
1030 IFL=ZZ+LR-8:MUSIC"F1"
1032 IF L=AR:MUSIC"F1":MUSIC"A1"
1033 IF RL$="n":GOTO1040
1035 IF L=ZZ+LR-2:GOTO1230
1040 GETA$
1050 IF L<=AR GOTO 1080
1060 IF A$=" " GOTO 1230
1070 IF A$="-":S$(J,ST)=S$(J,ST)+A$:GOTO 1230
1080 IF A$="" :IF R=0:GOTO 1040
1090 IF ASC(A$)=13:GOTO1230
1100 IF A$="£":GOTO 90
1110 IF A$="→":LT=LT+1:GOTO 2840
1120 IF A$="←":RT=RT+1:GOTO 2880
1130 IF A$="↓":GOTO 750
1140 IF A$="*":GOSUB 1320
1150 IF A$="\":IF R=0:R=1:GOTO 1170
1160 IF A$="\":IF R=1:R=0:GOTO 1040
1170 IF R=1:A$=RIGHT$(S$(J,ST),1)
1180 IF L<LR+ZZ:IF ASC(A$)>30:S$(J,ST)=S$(J,ST)+A$
1190 IF SIZE<100:PRINT"ARRET POUR RISQUE DE DEPASSEMENT DE CAPACITE MEMOIRE":GOT
O 1260
1200 IF LEN(S$(J,ST))>39:PRINTS$(J,ST);"■■":GOTO1220
1210 PRINT S$(J,ST);"_■■"
1220 L=L+1:GOTO1030
1230 IF J=ZS-5:MUSIC"A1";"A1"
1240 IF J=ZS-5:MUSIC"A1";"A1"
1250 NEXTJ
1260 PRINT"■■ VOUS DISPOSEZ ENCORE DE";SIZE;" OCTETS":PRINT"        APPUYER SUR
UNE TOUCHE!"
1270 GET A$:IF A$="":GOTO 1270
1280 GOTO90
1290 S$(J,ST)="":I=0:REM MARGE GAUCHE
1300 IF I>=LR:RETURN
1310 S$(J,ST)=S$(J,ST)+" ":I=I+1:GOTO1300
1320 A=1 :REM CORRECTION
1330 GET A$
1340 IF A$="■":IF RK=0:RK=1:GOTO 1390
1350 IF A$="■":IF RK=1:RK=0
1360 IF A$="■":C=1:IF P>0:A=A+1
1370 IF A$="■":C=-1:IF A>1:A=A-1
1380 IF A$="*":A$="":L=L-1:RETURN
1390 IF RK=1:IF P>0:IF A>1:A=A+C
1400 P=LEN(S$(J,ST))-A
1405 IFP<0THENP=0
1410 IF ASC(A$)=24:IF P+A<ZZ+LR:S$(J,ST)=LEFT$(S$(J,ST),P+1)+" "+RIGHT$(S$(J,ST)
,A-1):L=L+1:GOTO1440
1420 IF ASC(A$)=16:IF P>0:S$(J,ST)=LEFT$(S$(J,ST),P)+RIGHT$(S$(J,ST),A-1):PRINT"
                                          ■■":L=L-1:GOTO1440
1430 IF ASC(A$)>30:S$(J,ST)=LEFT$(S$(J,ST),P)+A$+RIGHT$(S$(J,ST),A-1):A=A-1:IF A
<1:A$="":L=L-1:RETURN
1440 IF LEN(S$(J,ST))>39:PRINTS$(J,ST);"■■":GOTO1460
1450 PRINT S$(J,ST);"_■■"
1460 IF P<=38 PRINT TAB(P);"■";"■":GOTO1330
1470 PRINTTAB(P-39);"■■■■":GOTO 1330
1480 PRINT"■ ***EDITION ECRAN ET RESTRUCTURATION***■"
1490 PRINT"1=EDITION ECRAN ET CORRECTION"
1500 PRINT"2=COUPURE AUTOMATQUE DES LIGNES"
1510 PRINT"3=EFFACEMENT DE LIGNES"
1520 PRINT"4=DECALAGE DE LIGNES"
1530 PRINT"5=RETOUR AU PROGRAMME PRINCIPAL"
1540 PRINT"■CHOISISSEZ UN NOMBRE S.V.P."
1550 ON ERROR GOTO3010
1560 INPUT MW
1570 ON ERROR GOTO 0
1580 ON MW GOTO 1590,1850,2040,2110,90
1590 PRINT"■" :REM UP.3.1.
1600 ON ERROR GOTO 3010
1610 INPUT"QUELLE PAGE?";ST:GOSUB 3070
1620 INPUT"DEPUIS LIGNE: ";N:GOSUB 3040
1630 ON ERROR GOTO 0
1640 E=ZS:S=1
1650 FOR J=N TO E STEP S
1660 PRINT"■PAGE ";ST;"  LIGNE ";J
1670 PRINTS$(J,ST)
1680 FOR I=1 TO 50
1690 GETA$
1700 IF ASC(A$)=13:GOTO1740
1710 IF A$="£":GOTO1480
1720 NEXT I,J
1730 GOTO 1600
1740 GET A$:IF A$="":GOTO 1740
1750 IF SIZE<100:PRINT"ARRET POUR RISQUE DE DEPASSEMENT DE CAPACITE MEMOIRE":GOT
O 1260
1760 IF 1>LEN(S$(J,ST)):GOSUB 1290
1770 IF A$="*":GOSUB 1320:GOTO 1740
1780 IF A$="£":GOTO 1480
1790 IF ASC(A$)=18:S=-1:E=1:N=J:GOTO 1650
1800 IF ASC(A$)<30:S=1:E=ZS:N=J:GOTO1650
1810 IF LEN(S$(J,ST))<=LR+ZZ:S$(J,ST)=S$(J,ST)+A$
1820 IF LEN(S$(J,ST))>39:PRINTS$(J,ST);"■■":GOTO1840
1830 PRINT S$(J,ST);"_■■"
```

```
1840 GOTO 1740
1850 PRINT"■":REM UP. 3.2.
1860 ON ERROR GOTO 3010
1870 INPUT"COMBIEN DE LIGNES PAR PAGE? ";N:GOSUB3040
1880 ON ERROR GOTO 0
1890 PRINT"ATTENDEZ S.V.P."
1900 ST=1:FOR J=1 TO ZS:S$(J,0)="":NEXTJ
1910 LX=ZS:LZ=ZS:GOSUB 1970:IF ST<19:ST=ST+1:GOTO 1910
1920 ST=0
1930 LX=ZS:LZ=ZS:GOSUB 1970:IF ST<19:ST=ST+1:GOTO 1930
1940 FOR J=1 TO ZS:S$(J,ST+1)=S$(J,ST):S$(J,ST)="":NEXTJ
1950 IF ST>0:ST=ST-1:GOTO1940
1960 GOTO 1480
1970 IF S$(LZ,ST)="":IF LZ>0:LZ=LZ-1:GOTO1970
1980 IF S$(LX,ST+1)="":IF LX>0:LX=LX-1:GOTO1980
1990 VZ=N-LZ:IF VZ<=0:RETURN
2000 IF LZ>LX:VZ=LX:IF VZ<=0:RETURN
2010 FOR J=1 TO VZ:S$(LZ+J,ST)=S$(J,ST+1):NEXT J
2020 FOR J=VZ+1 TO ZS:S$(J-VZ,ST+1)=S$(J,ST+1):NEXTJ
2030 FOR J=ZS-VZ TO ZS:S$(J,ST+1)="":NEXTJ:RETURN
2040 PRINT"■":REM UP.3.3.
2050 ON ERROR GOTO 3010
2060 INPUT"SUR QUELLE PAGE VOULEZ VOUS EFFACER?";ST:GOSUB 3070:SA=ST
2070 INPUT "PREMIERE LIGNE A EFFACER? ";N:GOSUB 3040:ZA=N
2080 INPUT "DERNIERE LIGNE A EFFACER? ";N:GOSUB 3040
2090 ON ERROR GOTO 0
2100 ZE=N:SB=0:ZB=1:N=ZE-ZA:GOTO2230
2110 PRINT"■LE DECALAGE DOIT S'EFFECTUER:          ":REM UP.3.4.
2120 ON ERROR GOTO 3010
2130 INPUT"DE LA PAGE";ST:GOSUB 3070:SA=ST
2140 INPUT"DE  LIGNE ";N:GOSUB 3040:ZA=N
2150 INPUT"A   LIGNE ";N:GOSUB 3040:ZE=N
2160 INPUT"A LA PAGE ";ST:GOSUB 3070:SB=ST
2170 INPUT"DE LIGNE ";N:GOSUB 3040:ZB=N
2180 ON ERROR GOTO 0
2190 N=ZE-ZA
2200 FOR J=ZS-N TO ZS
2210 IF S$(J,SB)<>"":PRINT"NOMBRE MAXIMUM DE LIGNES PAGE ";SB;"DEPASSE!":GOTO212
0
2220 NEXT J
2230 FOR J=ZS-N TO ZB+1 STEP-1:S$(J+N,SB)=S$(J-1,SB):NEXT J
2240 FOR J=ZA TO ZE:S$(ZB,SB)=S$(J,SA):ZB=ZB+1:NEXT J
2250 FOR J=ZE+1 TO ZS:S$(J-N-1,SA)=S$(J,SA):NEXT J
2260 FOR J=ZS-N TO ZS:S$(J,SA)="":NEXT J
2270 GOTO 1480
2280 PRINT"■     ***SAUVEGARDE SUR CASSETTE***■"
2290 ON ERROR GOTO 3010
2300 INPUT"QUELLE PAGE? ";ST:GOSUB 3070
2310 ON ERROR GOTO 0
2320 PRINT"■↓ RECORD/PLAY"
2330 WOPEN"PAGE"
2340 PRINT/T ST,ZS
2350 PRINT"PAGE";ST;" DE";ZS;" LIGNES EN COURS D'ECRITURE"
2360 FOR J=1 TO ZS:S$(J,ST)="*"+S$(J,ST):PRINT/T S$(J,ST):GOSUB 2550:NEXT J
2370 CLOSE:GOTO 90
2380 PRINT"■           ***LECTURE  CASSETTE***■"
2390 INPUT"CASSETTE DE TEXTE PRETE? (OUI=o) ";X$
2400 IF X$<>"o":GOTO 90
2410 PRINT"↓ PLAY"
2420 ROPEN"PAGE"
2430 INPUT/T ST,ZS
2440 PRINT"TROUVE: PAGE";ST;" DE";ZS;" LIGNES"
2450 INPUT"DOIT ELLE ETRE LUE? (OUI=o) ";X$
2460 IF X$<>"o":CLOSE:GOTO 90
2470 FOR J=1 TO ZS
2480 IF SIZE<300:CLOSE:GOTO 2520
2490 INPUT/T S$(J,ST)
2500 IF LEFT$(S$(J,ST),1)<>"*":S$(J-1,ST)=S$(J-1,ST)+",  "+S$(J,ST):J=J-1
2510 NEXT J
2520 FOR J=1 TO ZS:GOSUB 2550
2530 NEXTJ
2540 CLOSE:GOTO 90
2550 K=LEN(S$(J,ST))
2560 IF K>=1:K=K-1
2570 S$(J,ST)=RIGHT$(S$(J,ST),K)
2580 RETURN
2590 PRINT"■            ***IMPRESSION***■"
2600 ON ERROR GOTO 3010
2610 INPUT"QUELLE PAGE? ";ST
2620 GOSUB 3070
2630 INPUT"COMBIEN DE LIGNES A IMPRIMER? ";N
2640 GOSUB 3040
2650 ON ERROR GOTO 0
2670 FOR J=1 TO N
2680 PRINT/P S$(J,ST)
2690 NEXT J
2700 GOTO 90
2710 FOR I=4653 TO4676:P=PEEK(I):Q=PEEK(I+64):POKEI,Q:POKE(I+64),P:NEXTI
2720 DATA4645,89,4709,189,4646,90,4710,162,4686,94,4750,148,4753,255,4712,190,47
13,128,4648,91,4649,93,4647,64,4711,147,4749,192,4743,35,4744,36,4745,37
2730 FOR I=1 TO 17:READ P,W:POKE P,W:NEXT I
2740 END
2750 PRINT"■            ***MOTS-TYPES***■"
2760 PRINT"LONGUEUR DES MOTS: MAX.15 CARACTERES■"
2770 ON ERROR GOTO 3010
2780 INPUT"TOUCHE A DEFINIR? (1-10,0=RETOUR):";MW:IF MW>0:IF MW<=10:GOTO2800
2790 ON ERROR GOTO 0:GOTO90
2800 ON ERROR GOTO 0
2810 INPUT"■MOT-TYPE?                  ";X$
2820 DEF KEY (MW)=X$
2830 GOTO 2750
2840 IF LT>10 GOTO 1040:REM LINKSTAB
2850 IF LEN(S$(J,ST))>=LI(LT):LT=LT+1:GOTO2840
2860 FOR I=1 TO LI(LT)-LEN(S$(J,ST)):S$(J,ST)=S$(J,ST)+" ":L=L+1:NEXT I
2870 GOTO 1040
2880 IF RT>10 GOTO1040 :REM TAB.DROIT
2890 IF RE(RT)>LR+ZZ-L:RT=RT+1:GOTO 2880
2900 IF RE(RT)=0:GOTO1040
2910 H$=S$(J,ST)
2920 GOSUB3110
2930 M$=S$(J,ST)+M$
2940 M$=RIGHT$(M$,(LEN(M$)-LEN(H$)))
2950 AB=LR+ZZ-LEN(H$)-LEN(M$)-RE(RT) :REM DISTANCE
2960 IF AB<1:RT=RT+1:GOTO1040
2970 FOR I=1 TO AB:H$=H$+" ":NEXT I
2980 S$(J,ST)=H$+M$
2990 L=LEN(S$(J,ST))-1
3000 GOTO 1200
3010 PRINT"ERREUR D'ENTREE FORMELLE"
3020 PRINT"RECOMMENCER L'ENTREE!"
3030 RESUME
3040 IF N >=1:IF N<=ZS:RETURN
3045 IFN=0THEN90
3050 PRINT"70 LIGNES MAXIMUM!  NOMBRE RETENU: ";ZS
3060 INPUT"REPETER L'ENTREE ";N:GOTO3040
3070 IF ST>=1:IF ST<=20:RETURN
3075 IFST=0THEN90
3080 PRINT"LA MEMOIRE NE CONTIENT QUE LES PAGES 1-20"
3090 INPUT"PAGE ? ";ST
3100 GOTO 3070
3110 M$="":PRINT"■■■":REM UP. ENTREE
3120 GET A$:IF A$="":GOTO 3120
3130 IF SIZE<100:PRINT"ARRET POUR RISQUE DE DEPASSEMENT DE CAPACITE MEMOIRE":GOT
O 1260
3140 IF ASC(A$)=13:RETURN
3150 IF LEN(M$+S$(J,ST))>=ZZ+LR:RETURN
3160 IF A$="*":S$(J,ST)=S$(J,ST)+M$:M$="":GOSUB1320
3170 IF ASC(A$)>30:M$=M$+A$
3180 IF LEN(S$(J,ST)+M$)>39:PRINT S$(J,ST)+M$;"■■■":GOTO 3120
3190 PRINT S$(J,ST)+M$;"■■■":GOTO3120
```

# L'ACCES AUX ACCENTS

La programmation, en ROM, des caractères accentués sur vos différents MZ et imprimantes (bulletin n° 8) vous a semble-t-il posé beaucoup de problèmes ; essentiellement pour 2 raisons :

**1.** Les programmateurs d'EPROMS ne sont pas aussi répandus que les moulins à café et vous avez baucoup de mal à vous en faire prêter un.

**2.** Les bons programmateurs d'EPROMS sont encore plus rares, et si, miraculeusement, votre concierge a consenti (juste avant les étrennes...) à vous prêter le sien, les résultats obtenus n'ont pas tout à fait comblé vos espérances.

La solution ? Programmer ces mémoires rébarbatives au club. Alors, nous avons investi et nous pouvons maintenant vous proposer ce nouveau service.

D'abord quelques explications : Cette mémoire se présente, sur le circuit imprimé de votre MZ, sous la forme d'un circuit intégré muni de 14 pattes et surmonté d'une fenêtre transparente généralement obturée par un adhésif argenté. Cette mémoire conserve tous les caractères affichables à l'écran, d'où son nom : générateur de caractères. Au pied (aux pattes, si vous préférez) de ce boitier, sur le circuit imprimé, doit être inscrit « CG » : « Caractere generator » en anglais. Si toutes cesconditons sont réunies, bravo ! vous avez découvert ce que vous allez devoir changer.

Ensuite, pour obtenir, enfin, un véritable alphabet français sur votre MZ, il vous suffit de suivre la procédure suivante :

**1-** Procurez-vous, chez un marchand de composants, une EPROM 2716 monotention. Si vous nous envoyez une EPROM de récupération, assurez-vous qu'elle soit vierge, nous ne pourrions, autrement, vous la programmer. Enrobez-la de papier d'aluminium et envoyez l'ensemble au club, accompagné d'un chèque de 50 F.

**2-** Dès le retour de votre puce, démontez délicatement l'actuel générateur de caractères en repérant préalablement son sens (encoche), montez votre nouveau circuit à la place, tout aussi délicatement et DANS LE MEME SENS, refermez le tout et branchez.

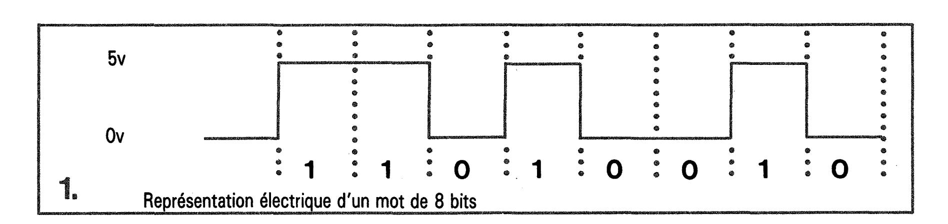
**3-** Si vous obtenez un écran tout blanc, c'est qu'il y a un défaut. SInon, appelez tous vos caractères graphiques et vous verrez enfin apparaître les é, è, ç, à, ê, ù, tant attendus. Il vous suffira, ensuite de taper le caractère graphique correspondant pour obtenir la lettre accentuée de votre choix.

Générateurs actuellement disponibles :
MZ 80K, IMPRIMANTE P3, MZ 80B, MZ 700

Nous recherchons avidement les générateurs suivants :
MZ 80A, IMPRIMANTE MZ80 P5/A, P5/B

S.B.

# LM SUR MZ




1. Représentation électrique d'un mot de 8 bits

**Pour poursuivre plus avant cette série d'articles avec un maximum de chances du succès nous allons devoir passer par une phase quelque peu rébarbative mais ABSOLUMENT INDISPENSABLE pour la compréhension de nos futurs articles. Il serait, en effet, illusoire de vouloir programmer en L.M. sans avoir comment fonctionne un microprocesseur et comment il interprête et manipule les informations qu'il est amené à gérer. C'est ce que nous allons tenter d'élucider.**

## L'INFORMATION

Toutes les tâches d'un microprocesseur se résument en 2 fonctions primaires : Recevoir et émettre des informations. D'autre part, nous avons vu que notre microprocesseur était relié au monde extérieur par une série de fils ; nos informations vont donc voyager sur les fils selon un codage particulier reconnaissable par notre microprocesseur et les autres-composants qui composent notre MZ.

Quel est le codage ? Sur nos fils, ne circulent que 2 tensions électriques différentes : 0 volts et 5 volts qui correspondent à 2 signaux différents que le microprocesseur va traduire, dans sa logique par 0 (0 volt) et 1 (5 volts). Pourquoi cette logique si simple ? Parce que le Z80, comme tous les microprocesseurs, travaille en BINAIRE ou BASE 2 et qu'il ne sait lire ou écrire que des 0 ou des 1. L'unité de base d'information de notre système ne peut donc avoir que 2 valeurs, elle s'appelle BIT (de BInary digiT) ; et c'est la succession de ces bits qui formera des « mots » transmis à l'écran, à la cassette, à l'imprimante.

## L'OCTET

Comme vous le savez déjà, le Z80 est un microprocesseur 8 BITS, c'est-à-dire qu'il ne peut recevoir ou émettre qu'un « mot » ou « nombre » composé d'une succession de 8 BITS ; c'est ce nombre, appelé OCTET que nous allons étudier.

Un octet

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| (128) | (64) | (32) | (16) | (8) | (4) | (2) | (1) |

## UN PEU D'ALGEBRE DE BOOLE

Dans 1 BIT, comme nous l'avons vu, nous ne pouvons loger que 2 valeurs : 0 ou 1 ; dans 2 bits successifs, nous pouvons donc loger 4 valeurs différentes (2 valeurs du 1° bit) * (2 valeurs du 2e bit), dans 3 bits : 8 valeurs (2*2*2)..... dans 8 bits : 256 valeurs. Nous voyons donc qu'il existe 256 possibilités différentes de combiner les 0 et les 1 des 8 bits qui composent notre octet.

Dans un octet, le bit de droite prend la valeur 0 ou 1, son voisin de gauche, la valeur 2 ou 0 ; le suivant 4 ou 0, etc, jusqu'au bit de gauche qui prend la valeur 128 ou 0. Ainsi, si l'on veut écrire un nombre quelconque, compris entre 0 et 255, en binaire, il suffit de raisonner ainsi :

**Par exemple,** le nombre 174.

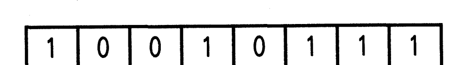
| | | | |
|---|---|---|---|
| 174 plus grand que | 128 donc : | bit 8 à 1. | 174 – 128 = 46 |
| 46 plus petit que | 64 donc : | bit 7 à 0. | |
| 46 plus grand que | 32 donc : | bit 6 à 1. | 46 – 32 = 14 |
| 14 plus petit que | 16 donc : | bit 5 à 0. | |
| 14 plus grand que | 8 donc : | bit 4 à 1. | 14 – 8 = 6 |
| 6 plus grand que | 4 donc : | bit 3 à 1. | 6 – 4 = 2 |
| 2 égale | 2 donc : | bit 2 à 1: | 2 – 2 = 0 |
| Résultat nul | donc : | Tous les bits (éventuels) suivants à 0. | |

Voilà, le résultat est donc :

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |

Pour effectuer le calcul inverse, il suffit d'additionner les valeurs relatives de chaque bit positionné à 1.

Exemple :



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |

donnera : 128 + 16 + 4 + 2 + 1 = 151

Pour terminer ce chapitre consacré au binaire, nous vous soumettons quelques valeurs binaires et décimales à convertir. Les solutions... dans le programme joint.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |

128, 255, 200, 74, 154

N'hésitez pas à faire. vous même, de nombreuses conversions de ce genre, vous maîtriserez ainsi mieux le calcul binaire et assimilerez plus rapidement certaines commandes du langage machine.




## L'HEXADECIMAL

Bien que cela soit tout à fait possible, il n'existe plus un seul programmeur assez fou pour programmer sa machine en binaire ; c'est très long, fastidieux, et source de nombreuses erreurs. Il a donc fallu inventer un langage intermédiaire, adapté au codage binaire et beaucoup moins rébarbatif pour l'homme ; ainsi est née la notation HEXADECIMALE.

Le but de cette notation est de pouvoir coder, avec 2 caractères les 256 valeurs possibles que peut contenir notre octet. Malheureusement, notre bonne vieille BASE 10 que nous utilisons depuis que nous avons 10 doigts est insuffisante pour coder ces valeurs puisque la valeur maximum obtenue avec 2 caractères est 99. Un rapide calcul nous conduit à la base 16, première capable de coder 256 valeurs avec 2 caractères.

Comment compte-t-on en BASE 16 ? eh bien, c'est très simple. Il suffit de considérer que nous avons 16 unités au lieu de 10 en BASE 10. Le codage de ces unités se fait de la manière suivante :

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| BASE 10 : | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| | 10 | 11 | 12 | 13 | 14 | 15 | | | | |
| BASE 16 : | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| | A | B | C | D | E | F | | | | |

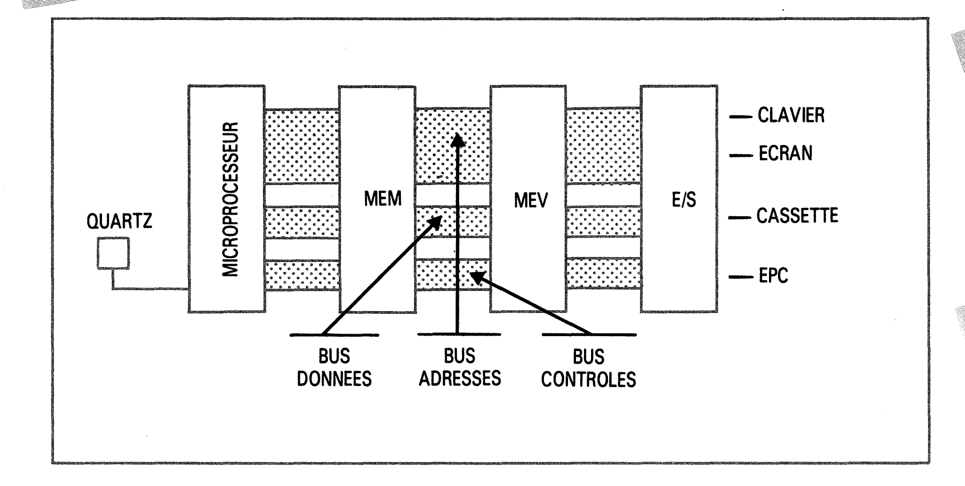
ainsi :

12 en base 10 devient OC en BASE 16
16 (( 1*10) + 6) devient 10 ((1*16) + 0)
24 (( 2*10) + 4) devient 18 ((1*16) + 8)
45 (( 4*10) + 5) devient 2D ((2*16) + D)
161 ((16*10) + 1) devient A1 ((A*16) + 1)
etc.

Il est très important de bien assimiler cette notation car tous les codes L.M. entrés ou lus sur la mémoire le sont en HEXA (diminutif de HEXADECIMAL). Vous pouvez voir une multitude de ces codes grace à la commande « D » du moniteur ; ils correspondent au contenu des mémoires lues.

En guise d'exercice, établissez votre propre table de conversion DECIMALE / HEXADECIMALE / BINAIRE. Ce document vous sera très utile pour vos futurs programmes L.M.

## UN PEU DE "HARD"




Le schéma ci-dessus vous représente une configuration informatique de base ; c'est-à-dire que tout système à microprocesseur est doté, au minimum, des éléments de ce schéma. Examinons-les, de gauche à droite :

• **LE QUARTZ** : C'est l'horloge du système qui émet très régulièrement 1 à 4 millions de battements par seconde et règle ainsi la vitesse de fonctionnement du microprocesseur et de certains autres composants périphériques.

• **LE MICROPROCESSEUR** : C'est le cœur du système ; il reçoit, calcule et émet des informations sur 8 bits ; nous l'avons déjà abordé dans notre précédent article, nous y reviendront longuement à l'avenir.

• **LA MEMOIRE MORTE (MEM)** : C'est un type de mémoire dans laquelle le microprocesseur peut lire des informations mais où il ne peut en écrire. En général, cette mémoire est programmée une fois pour toutes lors de la fabrication du micro-ordinateur, elle est inaltérable, ineffaçable par l'utilisateur et peut conserver son programme plusieurs siècles... Il est toutefois possible de l'effacer et de la réenregistrer avec un effaceur à ultra-violets et un programmateur d'EPROM. Cette mémoire peut également s'appeler ROM (de l'Anglais Read Only Memory), PROM ou EPROM.

Dans tous les MZ, la mémoire morte se trouve entre 0000 et 1000 H soit 4096D ; c'est le programme moniteur qui permet d'initialiser la machine à allumage et attend ensuite vos instructions.

• **LA MEMOIRE VIVE (MEV)** : Cette mémoire est totalement disponible au microprocesseur, il peut y lire et écrire des informations, les modifier, les effacer. C'est la mémoire sur laquelle vous chargez le BASIC ou autre langage, c'est elle, également qui supporte vos propres programmes et mémorise les caractères affichés sur votre écran.

• **LES ENTREES-SORTIES (E/S)** : Non, ne cherchez pas de sortie de secours pour octets en détresse sur votre MZ, il n'y en a pas. Les entrées-sorties servent simplement à gérer les différents accessoires périphériques au microprocesseur : Clavier, écran, cassette, imprimante, disquettes, poignées de jeux, etc., et à permettre une bonne entente au niveau des signaux transmis entre ces derniers et le microprocesseur qui les commande.

• **LES BUS** : (en Anglais : Reliable Automatic Transfer Pack). Les bus, ce sont les groupes de fils qui relient, entre eux les différents composants de la configuration.

3 BUS sont nécessaires à la configuration de base :

• **LE BUS DE DONNES** : Il est composé de 8 fils parallèles ; sur ces 8 fils, le microprocesseur peut recevoir ou émettre simultanément 8 tensions de 0 ou 5 volts, donc 8 bits. Comme le contenu de chacune des mémoires est codé sur



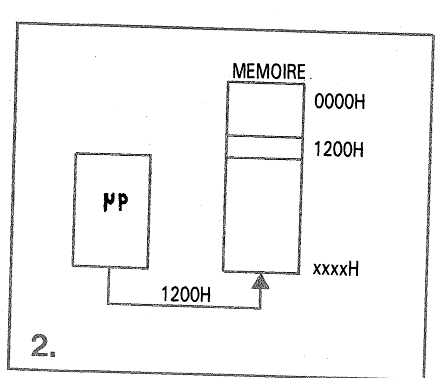

8 bits, le microprocesseur peut donc lire, ou écrire, en une seule opération (1 cycle), un mot de 8 bits en mémoire.

• **LE BUS D'ADRESSES** : Il est composé de 16 fils parallèles. Selon le même principe que le bus de données il supporte des mots de 16 bits.

Un rapide calcul nous apprend que sur 16 bits, nous pouvons coder 65 536 valeurs différentes. Les informations, sur ce bus, ne voyagent que dans un sens : émission du microprocesseur vers la mémoire. Le nombre envoyé sur ce bus (de 0 à 65 535) va servir à sélectionner une case mémoire (ou adresse mémoire) dans laquelle le microprocesseur désire lire ou écrire une information.

• **LE BUS DE CONTROLE** : Pour que le transfert de toutes ces informations sur les 24 fils que nous venons d'étudier se passe bien, il faut, quand même, un minimum de discipline ! C'est le rôle du bus de contrôle ; il gère et synchronise les transferts sur les bus de données et d'adresses.
Un bon exemple valant mieux que de longues explications, nous allons étudier une phase de lecture mémoire. Dans cet exemple, le microprocesseur va lire le contenu de l'adresse 1200H.

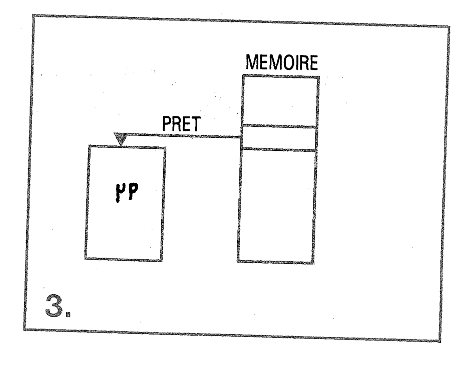
- **1er temps** : Le microprocesseur envoie sur les 16 fils de son bus d'adresses (appelés A0 à A15) le nombre binaire 1200H soit 0001001000000000. Les 0 correspondant à 0 volt et les 1, à 5 volts (fig. 2).



2.

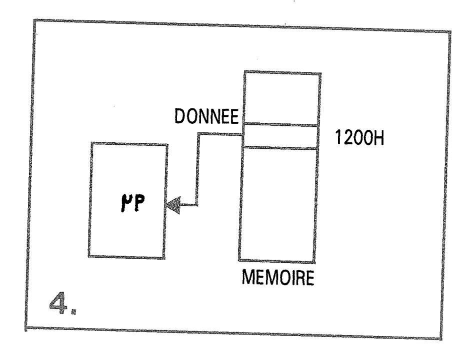
- **2e temps** : L'adresse numéro 1200H se reconnaît et envoie sur un fil du bus de contrôle un signal (5 V.) indiquant qu'elle est prête à dialoguer avec le microprocesseur (fig. 3).

- **3e temps** : Le microprocesseur envoie, sur un autre fil du bus de contrôle un signal indiquant qu'il veut lire le contenu de cette adresse.



3.

- **4e temps** : Le contenu binaire de l'adresse 1200H se place sur les 8 fils du bus de données et va se transporter jusqu'au microprocesseur qui va le ranger dans un de ses registres internes (fig. 4).



4.

Et voilà... E.L.E.M.E.N.T.A.I.R.E., mon cher Zilog. En fait, c'est un tout petit peu plus compliqué, mais le principe est le même.

Etudiez bien cet article et ses illustrations car (désolé, je me répète) il est indispensable pour la bonne compréhension de nos futurs articles.

En prime un petit programme de conversion adaptable sur tous MZ.

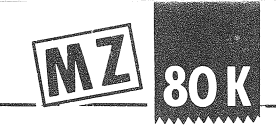
S. BIZOIRRE

**P.S.** : Peut-être avez-vous remarqué que :

R eliable
A utomatic
T ransfer
P ack

n'était pas une traduction tout à fait littérale du mot BUS... promis, nous ne recommencerons plus !

```
3 REM
4 REM CONVERTIONS BIN.DEC.HEXA.
5 REM
10 L$="
      "
20 PRINT"▣▣ Ce programme convertit en D
ECIMAL,"
30 PRINT"  HEXADECIMAL, et BINAIRE une v
aleur"
40 PRINT"  entree sous l'une de ces 3 fo
rmes."
50 PRINT"▣ Precisez 'D', 'H', ou 'B' ap
res la"
60 PRINT"  valeur. (Ex. 12537D)▣"
70 PRINT"▣TAPEZ UN TOUCHE."
80 GETRE$:IFRE$=""THEN80
90 PRINT"▣▣";L$
100 INPUT"▣ Valeur demandee : ";A$
110 C$=RIGHT$(A$,1):L=LEN(A$)
120 IF((C$="D")*(L>8))+((C$="H")*(L>7))+
((C$="B")*(L>19))THEN490
130 IF C$="D" THEN D$="9":GOSUB220:D=VAL
(LEFT$(A$,LEN(A$)-1)):GOTO160
140 IFC$="B"THEND$="1":GOSUB220:E=2:GOSU
B280:GOTO160
150 IFC$="H"THEND$="F":GOSUB220:E=16:GOS
UB280:GOTO160
160 E=2:GOSUB350:B$=E$:E=8:GOSUB350
170 T$=E$:E=16:GOSUB350:H$=E$
180 PRINT"▣ BINAIRE      : ";B$
190 PRINT"▣ DECIMAL      :";D
200 PRINT"▣ HEXADECIMAL  : ";H$;"▣"
210 GOTO 70
220 F=0:FOR I=1 TO LEN(A$)-1
230 B$=MID$(A$,I,1):IF B$<"0"THEN F=I
240 IF B$>D$ THEN F=I
250 NEXT I
260 IF F<>0 THEN 430
270 RETURN
280 D=0:FOR I=1 TO LEN(A$)-1
290 B$=MID$(A$,I,1)
300 IF B$>"9" THEN B=ASC(B$)-55:GOTO320
310 B=VAL(B$)
320 D=D*E+B
330 NEXTI
340 RETURN
350 G=D:E$=""
360 H=INT(G/E)
370 K=G-E*H
380 IFK>9 THEN K=K+55:F$=CHR$(K):GOTO400
390 F$=STR$(K)
400 E$=F$+E$
410 IFH<=0 THEN RETURN
420 G=H:GOTO 360
430 PRINT"▣Erreur de frappe sur le";F;
440 IF F=1THEN PRINT" er. ";
450 IF F>1THEN PRINT" eme. ";
460 PRINT" chiffre.▣"
470 FORI=1TO2500:NEXT
480 GOTO90
490 PRINT"▣ Trop long...
        ▣"
500 GOTO470
```



# PROGRAMMATION STRUCTURÉE

**Toujours dans la même optique que précédemment, nous allons voir un tri alphabétique programmé de deux façons différentes.**

**ALGORITHME (commun aux deux méthodes)**
Demander un nombre de mots. Générer X mots composés de 4 lettres tirées aléatoirement.

Afficher ces Xmots. Les classer par ordre alphabétique. Afficher de nouveau ces X mots classés. Afficher le temps mis pour ce classement. Afficher le nombre de tests, de permutations. Demander si on recommence.

## METHODE TRADITIONNELLE

```
10 CLR:PRINT"C"
20 INPUT"COMBIEN DE MOTS  ?";N
30 DIM A$(4):DIM B$(N)
40 FOR V=1 TO N
50 FOR K=1 TO 4
60 X=INT(RND(1)*91)
70 IF X<65 THEN60
80 A$(K)=CHR$(X)
90 NEXT K
100 B$(V)=A$(1)+A$(2)+A$(3)+A$(4)
110 PRINTB$(V);" ";
120 NEXT V
130 PRINT:PRINT".......PATIENCE....JE TRIE.........."
140 TI$="000000"
150 P=0:T=0
160 FOR B=1 TO N-1
170 FOR A=B+1 TO N
180 T=T+1
190 IF B$(A) < B$(B) THEN X$=B$(A):B$(A)=B$(B):B$(B)=X$:P=P+1
200 NEXT A,B
210 FOR I=1 TO N
220 PRINT B$(I);" ";
230 NEXT I
240 PRINT
250 PRINT"CLASSES EN ";MID$(TI$,3,2);" MIN ";
260 PRINTRIGHT$(TI$,2);" SEC."
270 PRINT"IL Y A EU ";T;" TESTS"
280 PRINT"IL Y A EU ";P;" PERMUTATIONS"
290 INPUT"ENCORE ?";R$
300 IF R$="O" THEN10
310 END
```

```
COMBIEN DE MOTS  ?48
LDXX KUKQ AUAL VEYR QRFR EGTM YZTY BWHZ
LWDE TZOL VDVP XXEW RKNY BBEP BTYV OVRM
LIGE BNMS KWFW VAZJ RTHE TXUP ZJYV TUNS
SQIP DPQZ AEXD FGLY NTFL CKPA WRJY RDFD
HQXY MXNP SHXZ PUVF DWYM AMKM QVVZ ARZQ
RWIY XMEX TYHA PQXH KUQX VKXW LVLF RWOM

.......PATIENCE....JE TRIE..........

AEXD AMKM ARZQ AUAL BBEP BNMS BTYV BWHZ
CKPA DPQZ DWYM EGTM FGLY HQXY KUKQ KUQX
KWFW LDXX LIGE LVLF LWDE MXNP NTFL OVRM
PQXH PUVF QRFR QVVZ RDFD RKNY RTHE RWIY
RWOM SHXZ SQIP TUNS TXUP TYHA TZOL VAZJ
VDVP VEYR VKXW WRJY XMEX XXEW YZTY ZJYV

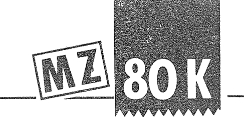
CLASSES EN 01 MIN. 22 SEC.
IL Y A EU  1128 TESTS
IL Y A EU  547 PERMUTATIONS
ENCORE ?N
READY
```

## METHODE STRUCTUREE

```
10  REPEAT
20        CLR:PRINT"C"
30        INPUT"COMBIEN DE MOTS  ?";N
40        DIM A$(4),B$(N)
50        K=0:V=0
60      REPEAT
70          REPEAT
80            X=INT(RND(1)*91)
90               WHILE X<65 DO
100                 X=INT(RND(1)*91)
110              WEND
120           A$(K)=CHR$(X)
130           K=K+1
140         UNTIL K=5
150       B$(V)=A$(1)+A$(2)+A$(3)+A$(4)
160       PRINTB$(V);" ";
170       K=0:V=V+1
180     UNTIL V=N
190        PRINT:PRINT".......PATIENCE..";
200        PRINT"..JE TRIE.........."
210        TI$="000000"
220        A=1:B=0:T=0:P=0
230             REPEAT
240                  REPEAT
250                    T=T+1
260                      WHILE B$(A) < B$(B)  DO
270                            P=P+1
280                            X$=B$(A)
290                            B$(A)=B$(B)
300                            B$(B)=X$
310                      WEND
320                    A=A+1
330                  UNTIL A=N
340               B=B+1:A=B+1
350             UNTIL B=N-1
360        PRINT
370                I=0
380                REPEAT
390                  PRINTB$(I);" ";
400                  I=I+1
410                UNTIL I=N
420        PRINT
430        PRINT"CLASSES EN ";
440        PRINT   MID$(TI$,3,2);" MIN. ";
450        PRINT   RIGHT$(TI$,2);" SEC. "
460        PRINT"IL Y A EU ";T;" TESTS"
470        PRINT"IL Y A EU ";P;" PERMUTATIONS"
480      INPUT"ENCORE ?";R$
490  UNTIL R$<>"O"
500 END
```

```
COMBIEN DE MOTS  ?48
ERDB BZAA YSVJ KPTO PUTO WQYL SBUE FWYV
FTEU WDEU JLEF KUII BRFN KAZZ LNEF FAKU
CZAE WPDX BIYU JDZB GRNT NZLK AONS ZMIW
YXOI PPGC RWHH BUQY REKU QLGR PZOB UMRL
PGXU MHLE SNHJ KNZR XDKW FAYY GHKS BFKV
CAYB AFMI UQWB NXOD PNIP WERG CGRK MIBX
.......PATIENCE....JE TRIE..........
AFMI AONS BFKV BIYU BRFN BUQY BZAA CAYB
CGRK CZAE ERDB FAKU FAYY FTEU FWYV GHKS
GRNT JDZB JLEF KAZZ KNZR KPTO KUII LNEF
MHLE MIBX NXOD NZLK PGXU PNIP PPGC PUTO
PZOB QLGR REKU RWHH SBUE SNHJ UMRL UQWB
WDEU WERG WPDX WQYL XDKW YSVJ YXOI ZMIW
```

```
CLASSES EN 00 MIN 47 SEC.
IL Y A EU  1128 TESTS
IL Y A EU  563 PERMUTATIONS
ENCORE ?N
READY
```



Pour les deux modes de programmation, le résultat est identique (sauf la vitesse du tri est légèrement différente, mais ce n'est pas l'objet de l'article).

## EXPLICATIONS POUR LA MÉTHODE STRUCTURÉE

Ligne 10 Début du pgm. Appairé avec UNTIL ligne n° 490

Ligne 20 Effacement des variables (ne sert à rien au premier RUN).

Ligne 30 Mise en mémoire de la valeur de N.

Ligne 40 Préparation d'un tableau (A$) de 5 cases (0,1,2,3,4) et, d'un autre (B$) de N cases.

Ligne 50 Mise à zéro des variables K et V. (inutile puisque CLR en 20, mais très bonne habitude).

Ligne 60 Début du S/pgm. qui produira N groupes de 4 lettres ; appairé avec UNTIL ligne n° 180.

Ligne 70 Début du S/S/pgm. qui produira 4 lettres.

Ligne 80 Tirage aléatoire (RND) d'un nombre entier (INT) entre Oet 91 (91 correspond à la lettre Z).

Ligne 90 Début d'un S/S/S/pgm. qui « tournera » TANT QUE le nombre (X) sera inférieur à 65 (65 correspond à la lettre A) appairé à WEND, ligne n° 110.

Ligne 120 La variable sera une lettre au premier tour (K=1), une autre au deuxième tour (K=2), etc. JUSQU'A CE QUE K=5 (ligne 140).

Ligne 130 Incrément du compteur k.

Ligne 150 La variable B$, (0) au premier tour est l'addition des 4 lettres. (Concaténation).

Ligne 160 Impression à l'écran de B$ (V).

Ligne 170 Remise à zéro de K. (important). Incrément du compteur V.

Ligne 180 Redémarrage du cycle, JUSQU'A CE QUE V soit égal à N (48 dans l'exemple).

Ligne 190 200, 210 sans commentaire !

Ligne 220 Provocation d'un décalage de 1 position entre les 2 premières chaînes de caractères. (A=1 B=0). T =compteur de tests. P =compteur de permutations.

Ligne 230 Démarrage du S/pgm. de l'ensemble du classement.

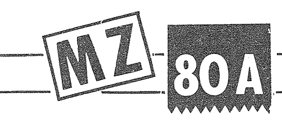
Ligne 240 Démarrage du S/S/pgm de test.

Ligne 250 Incrément du compteur test.

Ligne 260 TANT QUE B$ (1) est inférieur (0), (premier tour) FAIRE (DO) 1. Incrément du compteur permutation (ligne 270). 2. Permutation dans la mémoire de B$ (A) et de B$ (B) en passant par une variable intermédiaire (X$). Lignes 280, 290, 300).

Ligne 310 WEND, renvoit à la ligne 320. Incrément de B et de A (c'est-à-dire 2 et 1 pour le deuxième tour) et retour en ligne 230. Revenons à la ligne 260. Si B$ (1) est déjà inférieur à B$ (0) : saut tout de suite à la ligne 340.

La suite consiste à écrire, avec le même principe, les chaînes de caractères ainsi classées. Le SUPER BASIC 2 est doté de l'instruction EXC (échange). On peut remplacer les 3 lignes 280, 290, 300 par une seule : 290 ' EXC B$ (B). Cela évite de passer par une variable intermédiaire. (On gagne un peu de temps).
Dans le prochain bulletin, vous aurez 25 K. oct. de frappe. (programme structuré, bien sûr !).
A bientôt. Alain Bermond



**« Bien maigre, la rubrique du MZ80 A » nous reproche-t-on souvent ; « bien rare, votre courrier » serions-nous tentés de répondre. Les « A »istes seraient-ils jaloux de leurs découvertes au point de se les garder pour eux-seuls ?**
**Les astuces de ce numéro sont le fruit des découvertes de Philippe GAC : nous espérons que cet exemple sera suivi par beaucoup d'autres afin que votre rubrique conserve la place qu'elle mérite dans nos colonnes.**

## RECOPIE BASIC

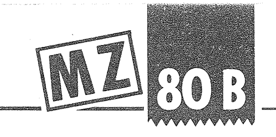
Selon la notice SHARP, il n'est pas possible d'obtenir de copies du BASIC d'après un BASIC déjà copié. Pour remédier à cette anomalie, il vous suffit de taper :
POKE 16895,49
POKE 16896,66
POKE 16919,0
puis appliquer la procédure du manuel concernant la copie du MASTER BASIC ; vous pourrez ainsi obtenir une multitude de copies ou recopies de votre BASIC même si vous ne possédez plus la cassette originale.

## QUELQUES DECOUVERTES

- Adresses des instructions BASIC et de leurs pointeurs :
- Début 5544 ($ 15A8)
- Fin 5944 ($ 1738)
- Début des lignes d'un programme BASIC : 20572 ($ 505C)
- Format des lignes d'un programme BASIC :

| Octet | Signification |
|---|---|
| 0-1 | Adresse (16 bits) du début de la prochaine ligne de texte BASIC |
| 2-3 | Numéro de la ligne de texte BASIC (16 bits) |
| 5 à X | Texte BASIC en ASCII sauf pour les instructions qui sont codées sur 1 octet (de 128 à 255). |
| X+* | Code retour chariot (CHR$(13) ou $0D) qui indique la fin de ligne BASIC. |

Une dernière routine intéressante : elle transforme le contenu du registre 16 bits HL en une valeur décimale ASCII rangée dans 5 octets à partir de l'adresse 19124 ; cette valeur se terminera par un $0D (retour chariot).
Exemple : si HL contient la valeur $1200, un CALL $1841 (c'est l'adresse de départ de cette routine) placera, à partir de l'adresse 19124 les codes ASCII « 4 », « 6 », « 0 », « 8 », « CR ». 4608 étant l'équivalent décimal de $1200 HEXA.



# DESASSEMBLEUR

**Le programme que nous vous proposons sera utile a ceux qui, lassés du BASIC, veulent tâter du langage machine. Après avoir entré le programme (bon courage), un menu général apparaît :**

## DESASSEMBLEUR :

donne les mnémoniques du langage machine (L.M. pour les intimes) sur l'écran ou sur imprimante. Les branchements relatifs sont indiqués en adresse absolue.

## CHARGEUR HEXA :

permet de créer un programme L.M. avec un minimum d'erreurs et dans un temps relativement court. Les codes opérations sont entrés en hexa et les branchements relatifs calculés par le programme.
On a la possibilité d'insérer un NOP (touche INST) ou de remplacer une instruction par des NOP (DEL).La touche HOME retourne à l'adresse initiale et les touches curseurs et permettent de se déplacer ligne a ligne (SPACE a la même action que ).
F1 permet d'entrer une chaîne de caractères en ASCII pour les titres L.M.
F5 exécute le programme qui est en cours de chargement (voir plus bas)
F10 met le pointeur sur une autre adresse.
**Remarque :** Le L.M. est toujours chargé à partir de C000, mais pour plus de commodité, on peut simuler une implantation différente. Les adresses sont recalculées.

## CONVERSION :

passage décimal hexa et réciproquement.

## RECHERCHE DE DEUX OCTETS :

donne les adresses d'implantation d'une succession de deux octets en mémoire.

## TRANSFERTS :

Le programme L.M., (a l'adresse C000 on vous l'a dit) peut être placé à une autre place en mémoire (RAM RAM) ou enregistré sur disquette (RAM FLOPPY) sous forme de code ASCII.
Il peut êgalment être lu sur la disquette (FLOPPY RAM).
Une petite modification des ignes 4720 a 4738 permettrait les transferts sur cassette.

## EXECUTION :

exécute le programme L.M. avec affichage du contenu des différents registres à la fin du travail.
Attention : tout L.M. doit être terminé par un RET (code C9) afin de redonner la main au BASIC. Sinon... à Dieu va... branchement au moniteur dans le meilleur des cas...

## VISIRAM :

affiche le contenu des mémoires en hexa et les oooooododdddddedes ASCII correspondants. En choisissant l'option DATE, on obtient le L.M. sous forme d'uATAS décimales (sur écran ou imprimante) prête à être réutilisée dans un autre programme pour charger le L.M. (READ A : POKE I,A). Arrêt du programme par « / »

## FIN :

C'est terminé : on quitte le désassembleur, mais le L.M. est toujours accessible par USR ($C000).

Jean MILLET

```
0 REM********************************
0 REM******* DESASSEMBLEUR **********
0 REM******** Jean Millet ***********
0 REM********** OCT 1984 ************
0 REM********************************
10 CONSOLEN,C40,S3,22:LIMIT$AFDF:S$=STRING$("-",40):CURSOR0,1:PRINTS$:CURSOR0,2
3:PRINTS$:HX$="0123456789ABCDEF":M0=45024:M1=49152:DEF KEY(1)=RUN100↓
20 DIMA0$(1),A1$(63),A2$(7),A3$(63),A4$(10),R$(7),AD(20):A0$(0)="(IX+":A0$(1)="
(IY+":M$="Patience, j'apprends les codes Z-80...":GOSUB600
30 FORI=0TO63:READA1$(I):NEXT:FORI=0TO7:READA2$(I):NEXT:FORI=0TO63:READA3$(I):N
EXT:FORI=0TO10:READA4$(I):NEXT:FORI=0TO7:READR$(I):NEXT
40 FORI=45056TO45109:READR:POKEI,R:NEXT:DEF FNP(X)=256*PEEK(X+1)+PEEK(X)
99 REM ============MENU==============
100 Y=2:GOSUB604:I=1:M$="* DESASSEMBLEUR Z-80 *":GOSUB602:PRINTTAB(7);STRING$("
*",25);:GOSUB150:PRINT"Desassembleur";:GOSUB150:PRINT"Chargeur hexa";:GOSUB150
110 PRINT"Conversion";:GOSUB150:PRINT"Recherche d'octets";:GOSUB150:PRINT"Trans
ferts";:GOSUB150:PRINT"Execution";:GOSUB150:PRINT"Visiram";
120 GOSUB150:PRINT"Fin";TAB(32);"*":PRINTTAB(7);"*";TAB(32);"*":PRINTTAB(7);STR
ING$("*",26):M$=SPACE$(30)+"Choix?":GOSUB600
130 GOSUB610:ONR-48GOTO4500,4000,4300,4400,4700,4900,4800,5000:GOTO130
150 PRINTTAB(32);"*":PRINTTAB(7);"*";TAB(32);"*":PRINTTAB(7);"*";I;" ";:I=I+1:R
ETURN
500 I=INT(N1/8):R=N1-8*I:RETURN
529 REM ============SUB. IMPLANTATION=
530 PRINT"⇓⇓⇓* IMPLANTATION";TAB(24);": ";H$;"H":PRINTTAB(26);:INPUT"⇑";H2$:H2$
=LEFT$(H2$,4):GOSUB2410:PRINTTAB(32);"⇑=";N2:RETURN
539 REM ============SUB. TITRE========
540 PRINT"®";TAB(9);"* TITRE  (16 CAR.MAX.) *":PRINT"⇓";TAB(13);STRING$("~",16)
:PRINT"⇑";TAB(13);:INPUT"";T$:IFLEN(T$)>16THEN540
545 FORI=1TO16:IFMID$(T$,I,1)<>"~"THENNEXT:RETURN
547 T$=LEFT$(T$,I-1):I=16:NEXT:RETURN
549 REM ============SUB. IMPRIMANTE===
550 MUSIC"C0":M$="EDITION SUR IMPRIMANTE ?  (O/N)⇐⇐⇐⇐⇐":GOSUB600:GOSUB610
552 IFR<>79THENP=0:PRINT"NON":LP=18:RETURN
555 P=1:MUSIC"D0":M$="Nombre de lignes du papier utilise :72":GOSUB600:CURSOR37
,24:INPUT"";L:LP=L-7:PRINT"®⇓⇓";TAB(11);"IMPRIMANTE S.V.P.!":GOSUB605:RETURN
599 REM ============SUB. AFFICHAGES===
600 Y=24:GOSUB604:CURSOR20-LEN(M$)/2,24:PRINTM$;:RETURN
602 Y=0:GOSUB604:CURSOR20-LEN(M$)/2,0:PRINTM$:Y=24:GOSUB604:PRINT"®";:RETURN
604 CURSOR0,Y:PRINTSPACE$(39);:RETURN
605 M$="Pour continuer, pressez une touche":GOSUB600:GOTO610
606 M$="FIN   Pressez une touche":GOSUB600:GOSUB610:GOTO100
607 PRINT"⇓⇓";TAB(12);"Diskette  S.V.P.!":GOTO605
609 REM ============SUB. REPONSE======
610 GETR$:IFR$=""THEN610
612 R=ASC(R$):RETURN
615 GOSUB610
616 R=R+7*(R>64)-48:IF(R<0)+(R>15)THEN615
617 RETURN
619 REM ============SAISIE ADDRESSE===
620 PRINT"⇓⇓* ADRESSE ";M$;TAB(24);": >...H⇐⇐⇐⇐⇐";:H2$="":N2=0:FORI=1TO4:GOSUB6
15:PRINTR$;">⇐";:H2$=H2$+R$:N2=R+16*N2:NEXT:PRINT"H =";N2:RETURN
625 PRINT"®";:M$="DEBUT":GOSUB620:D=N2:H$=H2$:M$="FIN":GOSUB620:F=N2
626 IFF<DTHEN625
627 RETURN
629 REM ============SAISIE OCTETS=====
630 J=0:S$=""
632 H1$="":N1=0:FORI=1TO2:PRINT">⇐";
635 GOSUB610:IF(R=7)*(J>0)*(I=1)THENJ=J-1:A=A-1:PRINT" ⇐⇐⇐⇐     ⇐⇐⇐⇐>⇐";:GOTO632
640 GOSUB616:PRINTR$;:H1$=H1$+R$:N1=R+16*N1:NEXT:POKEA,N1:A=A+1:S$=H1$+S$:PRINT
" ";:J=J+1:IFJ<NBTHEN632
645 RETURN
649 REM ============DESASSEMBLEUR=====
700 N2=A-DE:N2=N2-65535*(N2<0):GOSUB2310:L0$=H2$
705 L1$="":L2$="":LP$="":O=A:GOSUB2500:N=1:IFN1<64THENH$=A1$(N1):GOTO840
710 IFN1=118THENH$="HALT":GOTO1790
720 IFN1<128THENN1=N1-64:GOSUB500:H$="LD "+R$(I)+","+R$(R):GOTO1790
730 IFN1<192THENN1=N1-128:GOSUB500:H$=A2$(I)+R$(R):GOTO1790
800 H$=A3$(N1-192)
840 N=VAL(LEFT$(H$,1)):L=LEN(H$):IFL>1THENH$=RIGHT$(H$,L-1)
850 ONNGOTO1790,1060,1140,1240,1310,1520,1310:STOP
1050 REM===========CAS 2 OCTETS======
1060 O=A+1:GOSUB2500:S$=H1$+"H":GOSUB2520:GOTO1790
1130 REM===========CAS 3 OCTETS======
1140 O=A+1:GOSUB2500:S$=H1$+"H":O=A+2:GOSUB2500:S$=H1$+S$:GOSUB2520:GOTO1790
1230 REM============INST CBXX=========
1240 O=A+1:GOSUB2500:GOSUB500:N=2:IFI<8THENH$=A4$(I)+R$(R):GOTO1790
1250 I1=INT(I/8):R1=I-8*I1:H$=A4$(I1+7)+STR$(R1)+","+R$(R):GOTO1790
1300 REM============INST DDXX-FDXX====
1310 V=-(N=7):O=A+1:GOSUB2500:IFH1$="CB"THEN1660
1330 RESTORE3400:FORM=1TO39:READR$:IFH1$=LEFT$(R$,2)THEN1370
1350 NEXTM:GOTO1760
1370 N=VAL(MID$(R$,3,1))+1:H$=RIGHT$(R$,LEN(R$)-3):IFV=1GOSUB2730
1410 IFN=2THEN1790
1420 O=A+2:GOSUB2500:S$=H1$+"H":IFN=3THENGOSUB2520:GOTO1790
1450 O=A+3:GOSUB2500:IFM<>11THENS$=H1$+S$:GOSUB2520:GOTO1790
1470 GOSUB2520:S$=H1$+"H":GOSUB2520:GOTO1790
1510 REM============INST ED===========
1520 O=A+1:GOSUB2500:RESTORE3500:FORM=1TO56:READR$:IFH1$=LEFT$(R$,2)THEN1570
1550 NEXTM:GOTO1760
1570 N=VAL(MID$(R$,3,1))+1:H$=RIGHT$(R$,LEN(R$)-3):IFN=2THEN1790
1600 O=A+2:GOSUB2500:S$=H1$+"H":O=A+3:GOSUB2500:S$=H1$+S$:GOSUB2520:GOTO1790
1640 REM============INST DDCB-FDCB====
1660 O=A+2:GOSUB2500:S$=H1$+"H)":O=A+3:GOSUB2500:GOSUB500:IFR<>6THEN1760
1670 IFI<8THENH1$=A4$(I):GOTO1700
1680 I1=INT(I/8):R1=I-8*I1:H1$=A4$(I1+8)+R$(R1)
1700 H$=H1$+A0$(V)+S$:N=4:GOTO1790
1740 REM============INST INVALIDE=====
1760 LP$="":L1$="":L2$="":O=A:GOSUB2500:L3$="??..":L4$="":L5$="":N=1:RETURN
1780 REM============FIN DE DESAS.=====
1790 L=LEN(H$):FORI=1TOL:IFMID$(H$,I,1)<>" "THENNEXTI:L3$=H$:L4$="":GOTO1810
1800 L3$=LEFT$(H$,I-1):L4$=RIGHT$(H$,L-I)
1810 L5$="":IFL3$="CALL"THENL5$="==":RETURN
1820 IF(L3$="JR")+(L3$="DJNZ")THENS=PEEK(A+1):N2=A+2-DE+S+256*(S>127):GOSUB2310
:L5$="→"+H2$+"H":RETURN
1830 RETURN
2250 REM==CONV DECI-HEXA COURT========
2260 Z1=INT(N1/16):Z0=N1-Z1*16:H1$=MID$(HX$,Z1+1,1)+MID$(HX$,Z0+1,1):RETURN
2310 REM==CONV DECI-HEXA LONG=========
2320 N1=INT(N2/256):GOSUB2260:H2$=H1$:N1=N2-256*N1:GOSUB2260:H2$=H2$+H1$:RETURN
2400 REM==CONV HEXA-DECI LONG=========
2410 N2=0:FORI=1TO4:Z=ASC(MID$(H2$,I,1))-48:Z=Z+(Z>9)*7:N2=Z+16*N2:NEXTI:RETURN
```

```
 2499 REM==CREATION L1$ ET L2$=========
 2500 N1=PEEK(0):GOSUB2260:C$=" ":IF(N1>32)*(N1<256)THENC$=CHR$(N1)
 2510 L2$=L2$+C$:L1$=L1$+H1$:LP$=LP$+H1$+" ":RETURN
 2520 REM==INSERTION EN H$ DE S$=======
 2530 L=LEN(H$):FORI=1TOL:IFMID$(H$,I,1)<>"↑"THENNEXT:GOTO1760
 2540 H$=LEFT$(H$,I-1)+S$+RIGHT$(H$,L-I):I=L:NEXT:RETURN
 2710 REM ===REMPL. DE X PAR Y DANS H$=
 2730 L=LEN(H$):FORI=3TOL:IFMID$(H$,I,1)<>"X"THENNEXT:GOTO1760
 2740 H$=LEFT$(H$,I-1)+"Y"+RIGHT$(H$,L-I):I=L:NEXT:RETURN
 2800 REM==CODES < 40 ================
 2810 DATA"1NOP","3LD BC,↑","1LD (BC),A","1INC BC","1INC B","1DEC B","2LD B,↑","
1RLCA","1EX AF,AF'","1ADD HL,BC","1LD A,(BC)","1DEC BC","1INC C","1DEC C"
 2820 DATA"2LD C,↑","1RRCA","2DJNZ ↑","3LD DE,↑","1LD (DE),A","1INC DE","1INC D"
,"1DEC D","2LD D,↑","1RLA","2JR ↑","1ADD HL,DE","1LD A,(DE)","1DEC DE","1INC E"
 2830 DATA"1DEC E","2LD E,↑","1RRA","2JR NZ,↑","3LD HL,↑","3LD (↑),HL","1INC HL"
,"1INC H","1DEC H","2LD H,↑","1DAA","2JR Z,↑","1ADD HL,HL","3LD HL,(↑)"
 2840 DATA"1DEC HL","1INC L","1DEC L","2LD L,↑","1CPL","2JR NC,↑","3LD SP,↑","3L
D (↑),A","1INC SP","1INC (HL)","1DEC (HL)","2LD (HL),↑","1SCF","2JR C,↑"
 2850 DATA"1ADD HL,SP","3LD A,(↑)","1DEC SP","1INC A","1DEC A","2LD A,↑","1CCF"
 2999 REM==CODES < BF ET > 7F=========
 3000 DATA"ADD A,","ADC A,","SUB ","SBC A,","AND ","XOR ","OR ","CP "
 3179 REM==CODES > BF=================
 3280 DATA"1RET NZ","1POP BC","3JP NZ,↑","3JP ↑","3CALL NZ,↑","1PUSH BC"
 3290 DATA"2ADD A,↑","1RST 00H","1RET Z","1RET","3JP Z,↑","4","3CALL Z,↑","3CALL
↑","2ADC A,↑","1RST 08H","1RET NC","1POP DE","3JP NC,↑","2OUT (↑),A"
 3300 DATA"3CALL NC,↑","1PUSH DE","2SUB ↑","1RST 10H","1RET C","1EXX","3JP C,↑",
"2IN A,(↑)","3CALL C,↑","5","2SBC A,↑","1RST 18H","1RET PO","1POP HL"
 3310 DATA"3JP PO,↑","1EX (SP),HL","3CALL PO,↑","1PUSH HL","2AND ↑","1RST 20H","
1RET PE","1JP (HL)","3JP PE,↑","1EX DE,HL","3CALL PE,↑","6","2XOR ↑","1RST 28H"
 3320 DATA"1RET P","1POP AF","3JP P,↑","1DI","3CALL P,↑","1PUSH AF","2OR ↑","1RS
T 30H","1RET M","1LD SP,HL","3JP M,↑","1EI","3CALL M,↑","7","2CP ↑","1RST 38H"
 3329 REM==CODES CB==================
 3330 DATA"RLC ","RRC ","RL ","RR ","SLA ","SRA ","","SRL ","BIT ","RES ","SET "
 3339 REM==MNEMO REGISTRES============
 3340 DATA"B","C","D","E","H","L","(HL)","A"
 3349 REM==PG. MACHINE================
 3350 DATA205,0,192,50,224,175,237,67,225,175,237,83,227,175,34,229,175,237,115,
231,175,221,34,233,175,253,34,235,175,229,245,237,87,50,237,175,237,95,50,238
 3360 DATA175,225,34,239,175,229,241,225,227,34,241,175,227,201
 3399 REM==GROUPE DDXX FDXX  XXCB======
 3400 DATA"091ADD IX,BC","191ADD IX,DE","213LD IX,↑","223LD (↑),IX","231INC IX",
"291ADD IX,IX","2A3LD IX,(↑)","2B1DEC IX","342INC (IX+↑)","352DEC (IX+↑)"
 3410 DATA"363LD (IX+↑),↑","391ADD IX,SP","462LD B,(IX+↑)","4E2LD C,(IX+↑)","562
LD D,(IX+↑)","5E2LD E,(IX+↑)","662LD H,(IX+↑)","6E2LD L,(IX+↑)"
 3420 DATA"712LD (IX+↑),C","722LD (IX+↑),D","732LD (IX+↑),E","742LD (IX+↑),H","7
52LD (IX+↑),L","772LD (IX+↑),A","7E2LD A,(IX+↑)","862ADD A,(IX+↑),A"
 3430 DATA"8E2ADC A,(IX+↑)","962SUB (IX+↑)","9E2SBC A,(IX+↑)","A62AND (IX+↑)","A
E2XOR (IX+↑)","B62OR (IX+↑)","BE2CP (IX+↑)","E31EX (SP),IX","E51PUSH IX"
 3440 DATA"E91JP (IX)","F91LD SP,IX","702LD (IX+↑),B","E11POP IX"
 3499 REM==TABLE DES EDXX==============
 3500 DATA"401IN B,(C)","411OUT (C),B","421SBC HL,BC","433LD (↑),BC","441NEG","4
51RET N","461IM 0","471LD I,A","481IN C,(C)","491OUT (C),C","4A1ADC HL,BC"
 3510 DATA"4B3LD BC,(↑)","4D1RETI","501IN D,(C)","511OUT (C),D","521SBC HL,DE","
533LD (↑),DE","561IM 1","571LD A,I","581IN E,(C)","591OUT (C),E","5A1ADC HL,DE"
 3520 DATA"5B3LD DE,(↑)","5E1IM 2","601IN H,(C)","611OUT (C),H","621SBC HL,HL","
671RRD","681IN L,(C)","691OUT (C),L","6A1ADC HL,HL","6F1RLD","721SBC HL,SP"
 3530 DATA"733LD (↑),SP","781IN A,(C)","791OUT (C),A","7A1ADC HL,SP","7B3LD SP,(
↑)","A01LDI","A11CPI","A21INI",A31OUTI",A81LDD","A91CPD","AA1IND","AB1OUTD"
 3540 DATA"B01LDIR","B11CPIR","B21INIR","B31OTIR","B81LDDR","B91CPDR","BA1INDR",
"BB1OTDR","4F1LD R,A","5F1LD A,R"
 3999 REM========CHARGEUR HEXA=========
 4000 M$="* CHARGEUR HEXA *":GOSUB602:M$="Adresse en memoire :C000H = 49152":GOS
UB600:PRINT"®";:M$="IMPLANTATION":GOSUB620:DE=M1-N2:A=M1:AM=M1
 4001 M$="CR:Fin F1:ASC F5:Exec F10:Saut":GOSUB600:PRINT"®";:MO=0
 4002 A0=A:N2=A-DE:AM=AM+(AM-A)*(A>AM):GOSUB2320:PRINTH2$;">⇐";
 4003 GOSUB610:IFR=13THEN100
 4004 IFR=16THENMO=-(MO=0):GOTO4220
 4005 IF(R=1)+(R=32)THENGOSUB700:PRINT"-";TAB(8);LP$;TAB(21);L3$;" ";L4$;TAB(34)
;L5$:A=A+N:GOTO4002
 4006 IF(R=2)*(CSRV>3)THENCV=-1:GOSUB4200:A=N2+DE:PRINT" ";:CURSOR4,CSRV-1:PRINT
SPACE$(35);:CURSOR0,CSRV:GOTO4002
 4007 IFR=8THENFORI=AMTOASTEP-1:POKEI+1,PEEK(I):NEXT:POKEA,0:GOTO4205
 4008 IF(R=7)*(CSRV<22)THENCV=1:GOSUB4200:A1=N2+DE:FORI=ATOA1-1:POKEI,0:NEXT:GOT
O4205
 4009 IFR=5THENA=M1:PRINT"®";:GOTO4002
 4010 IFR=25THENPRINT"®":M$="SUIVANTE":GOSUB620:A=N2+DE:PRINT"®";:GOTO4002
 4011 IFR=20THENFR=0:N2=49152:GOTO4910
 4012 R=R+7*(R>64)-48:IF(R<0)+(R>15)THENUSR($EBE):GOTO4003
 4014 PRINT"-  ";R$;">⇐";:N1=16*R:GOSUB615:PRINTR$;" ";:N1=N1+R:POKEA,N1:A=A+1:I
FN1<64THENHC$=A1$(N1):GOTO4100
 4018 IFN1=118THENHC$="1HALT":GOTO4100
 4020 IFN1<128THENN1=N1-64:GOSUB500:HC$="1LD "+R$(I)+","+R$(R):GOTO4100
 4022 IFN1<192THENN1=N1-128:GOSUB500:HC$="1"+A2$(I)+R$(R):GOTO4100
 4024 REM==CODE CBXX
 4025 IFN1=203THENNB=1:GOSUB630:GOTO4150
 4026 REM==CODE DD OU FD
 4027 IF(N1=221)+(N1=253)THENV=-(N1=253):NB=1:GOSUB630:GOTO4160
 4029 REM==CODE ED
 4030 IFN1=237THENNB=1:GOSUB630:GOTO4170
 4035 HC$=A3$(N1-192)
 4100 NB=VAL(LEFT$(HC$,1))-1:H$=RIGHT$(HC$,LEN(HC$)-1):CS=CSRH:CR=CSRV:IF(LEFT$(
H$,2)<>"DJ")*(LEFT$(H$,2)<>"JR")THEN4108
 4102 PRINT" ";H$;"⇐  absolu : >...H⇐⇐⇐⇐⇐";:H2$="":N2=0:FORI=1TO4:GOSUB615:PRINT
R$;">⇐";:H2$=H2$+R$:N2=R+16*N2:NEXT:CURSOR10,CR:PRINTSPACE$(25);
 4106 N1=N2-A+DE-1:N1=N1-256*(N1<0):POKEA,N1:A=A+1:GOSUB2260:CURSORCS,CR:PRINTH1
$;:S$=H1$:GOTO4115
 4108 CURSOR20,CR:PRINTH$;:IFNB=0THENPRINT:GOTO4002
 4110 CURSORCS,CR:GOSUB630
 4115 S$=S$+"H":GOSUB2530:CURSOR20,CSRV:PRINTH$:GOTO4002
 4150 GOSUB500:IFI<8THENHC$="1"+A4$(I)+R$(R):GOTO4100
 4152 I1=INT(I/8):R1=I-8*I1:HC$="1"+A4$(I1+7)+STR$(R1)+","+R$(R):GOTO4100
 4160 IFN1=203THEN4180
 4162 RESTORE3400:FORM=1TO39:READR$:IFS$=LEFT$(R$,2)THEN4165
 4163 NEXTM:GOTO4195
 4165 H$=RIGHT$(R$,LEN(R$)-2):IFV=1THENGOSUB2730
 4166 IFS$="36"THENNB=1:GOSUB630:S$=S$+"H":GOSUB2530:HC$="2"+RIGHT$(H$,LEN(H$)-1
):GOTO4100
 4167 HC$=H$:GOTO4100
 4170 RESTORE3500:FORM=1TO56:READR$:IFS$=LEFT$(R$,2)THEN4172
 4171 NEXT:GOTO4195
 4172 HC$=RIGHT$(R$,LEN(R$)-2):GOTO4100
 4180 NB=1:GOSUB630:H$=S$+"H)":GOSUB630:GOSUB500:IF(R<>6)+(I=6)THEN4195
 4182 IFI<8THENHC$="1"+A4$(I):GOTO4185
 4183 I1=INT(I/8):R1=I-8*I1:HC$="1"+A4$(I1+7)+STR$(R1)+","
 4185 HC$=HC$+A0$(V)+H$:GOTO4100
 4195 CURSOR20,CSRV:PRINT".......???":A=A0:GOTO4002
 4200 H2$="":FORI=0TO4:H2$=H2$+CHARACTER$(I,CSRV+CV):NEXT:GOSUB2410:RETURN
 4205 PRINT"-";TAB(8);"00";TAB(21);"NOP":A=A+1:GOTO4002
 4220 PV=CSRV:PH=CSRH:IFMO=0THENCURSOR22,0:PRINT"HEXA *":CURSORPH,PV:GOTO4002
 4222 CURSOR22,0:PRINT"ASCI *":CURSORPH,PV:PRINT"-   →⇐";
 4225 GOSUB610:POKEA,R:A=A+1:IFR=13THENMO=0:PRINT"↵":GOTO4220
 4227 PRINTR$;"→⇐";:GOTO4225
 4299 REM========CONVERSIONS===========
 4300 M$="* CONVERSION *":GOSUB602:M$="Hexa→deci  Deci→hexa  Fin":GOSUB600
 4310 GOSUB610:PRINT"®":ON-(R=72)-2*(R=68)-3*(R=70)GOTO4320,4330,100:GOTO4310
 4320 CURSOR0,10:M$="HEXA":GOSUB620:GOTO4310
 4330 CURSOR0,12:PRINT"* VALEUR DECIMALE";TAB(24);:INPUT": ";N2:GOSUB2320:PRINTT
AB(32);"⇑= ";H2$;"H":GOTO4310
 4399 REM========RECHERCHE DE 2 OCTETS=
 4400 M$="*RECHERCHE D'UNE SEQUENCE DE 2 OCTETS*":GOSUB602:GOSUB625:M$="OCTETS :
":GOSUB600:CURSOR26,24:A=MO:NB=2:GOSUB630
 4420 PRINT"®":O1=PEEK(MO):O2=PEEK(MO+1):FORN2=DTOF:IFPEEK(N2)<>O1THEN4450
 4430 IFPEEK(N2+1)<>O2THEN4450
 4440 GOSUB2320:PRINTTAB(5);H2$;"H =";N2:MUSIC"+A0"
 4450 NEXT:GOTO606
 4499 REM========DESASSEMBLEUR=========
 4500 M$="* DESASSEMBLEUR Z80 *":GOSUB602:GOSUB625:GOSUB530:DE=D-N2:GOSUB540:A=D
 4510 GOSUB550:PRINT"®";:Y=0:GOSUB604:CURSOR0,0:PRINT"=PAGE=";TAB(12);T$;TAB(32)
;"DES-Z80":CURSOR0,2:PRINT"ADD. INSTRUC. ASCII  MNEMONIQUE OPER.":PA=0:Z$="---"
 4520 IFP=1THENPAGE/PLP+10
 4530 PA=PA+1:CURSOR7,0:PRINTPA:Y=24:GOSUB604:PRINT"®":LI=1:AD(PA)=A
 4540 IFP=1THENPRINT/P"=PAGE=";PA;TAB(14);T$;TAB(38);"DESAS-Z80":PRINT/PSTRING$(
"=",49):PRINT/P"ADDR.   INSTRUC.     ASCII  MNEMONIQUE OPER.":PRINT/P
 4550 GOSUB700:PRINTLO$;" ";L1$;TAB(14);L2$;TAB(19);L3$;TAB(24);L4$;TAB(33);L5$:
IFP=1THENPRINT/PLO$;"-    ";LP$;TAB(21);L2$;TAB(28);L3$;TAB(33);L4$;TAB(44);L5$
 4560 LI=LI+1:IFL1$="C9"THENPRINTTAB(19);Z$:LI=LI+1:IFP=1THENPRINT/PTAB(28);Z$
 4570 S$=LEFT$(L1$,2):IF(S$="18")+(S$="C3")+(S$="E9")THENPRINTTAB(19);STRING$("—
",10):LI=LI+1:IFP=1THENPRINT/PTAB(27);STRING$("—",12)
 4580 A=A+N:IFA>FTHEN606
 4585 IFLI<LPTHEN4550
 4587 IFP=1THENPRINT/PTAB(44);".../...";CHR$(5):GOTO4530
 4590 M$=".../...  Fin  Page predecente  Suite":GOSUB600:GOSUB610:ON-(R=70)-2*(R
=83)-2*(R=32)-3*(R=80)GOTO100,4530,4595:GOTO4590
 4595 PA=-(PA-2)*(PA>1):A=AD(PA+1):GOTO4530
 4699 REM========TRANSFERTS============
 4700 M$="* TRANSFERTS *":GOSUB602:M$="RAM→RAM  RAM→FLOPPY  FLOPPY→RAM":GOSUB600
:GOSUB610:Y=24:GOSUB604:ON-(R=82)-2*(R=77)-3*(R=70)GOTO4710,4720,4730:GOTO4700
 4710 GOSUB625:GOSUB530:FORI=DTOF:POKEN2-D+I,PEEK(I):NEXT:GOTO606
 4720 GOSUB540:GOSUB625:NB=F-D:GOSUB530:GOSUB607:PRINT"®⇓⇓";NB;" octets":WOPEN#1
,T$:PRINT#1,N2,NB:FORI=0TONB:PRINT#1STR$(PEEK(D+I)):NEXT:CLOSE#1:GOTO606
 4730 GOSUB540:GOSUB607:ROPEN#1,T$:INPUT#1,N2,NB:CLOSE#1:GOSUB2320:H$=H2$
 4735 PRINT"®⇓⇓";:GOSUB530:IFN2+NB>65535THENPRINT"⇓⇓Programme trop long :";NB;"
octets":GOSUB605:GOTO4735
 4738 ROPEN#1,T$:INPUT#1,X,X:FORI=0TONB:INPUT#1R$:POKEN2+I,VAL(R$):NEXT:CLOSE#1:
GOTO606
 4799 REM========VISIRAM===============
 4800 M$="* VISIRAM *":GOSUB602:GOSUB625:O=D:M$="ASC   DATA":GOSUB600:GOSUB610:O
N-(R=65)-2*(R=68)GOTO4810,4850:GOTO4800
 4810 V=24:GOSUB604:PRINT"®";:FORJ=1TO19:LP$="":L1$="":L2$="":N2=0:GOSUB2320:PRI
NTH2$;" ‖";:FORI=0TO7:GOSUB2500:O=O+1:GETR$:IF(O>F)+(R$="/")THENI=7:J=19
 4820 NEXTI:PRINTLP$;"⇐‖";TAB(32);L2$:NEXTJ:IF(O>F)+(R$="/")THEN606
 4830 GOSUB605:IFR$="/"THEN606
 4840 GOTO4810
 4850 V=24:GOSUB604:GOSUB550:PRINT"®";"⇓⇓DATA ";:FORI=DTOF:PRINTSTR$(PEEK(I));",
";:IFP=1THENPRINT/PSTR$(PEEK(I));",";
 4860 NEXT:IFP=1THENPRINT/P
 4865 GOTO606
 4899 REM =======EXECUTION=====
 4900 M$="* EXECUTION *":GOSUB602:H$="C000":GOSUB530:FR=1
 4910 X=INT(N2/256):POKE$B002,X:X=N2-256*X:POKE$B001,X
 4920 USR($B000):X=MO:N1=PEEK(X):GOSUB2260:PRINT"®⇓⇓⇓A =";H1$,:N2=FNP(X+1):GOSUB
2320:PRINT"BC=";H2$,:N2=FNP(X+3):GOSUB2320:PRINT"DE=";H2$,
 4930 N2=FNP(X+5):GOSUB2320:PRINT" HL =";H2$:PRINT:N2=FNP(X+7):GOSUB2320:PRINT"S
P=";H2$,:N2=FNP(X+9):GOSUB2320:PRINT"IX=";H2$,
 4940 N2=FNP(X+11):GOSUB2320:PRINT"IY=";H2$,:N2=PEEK(X+17):GOSUB2320:PRINT"(SP)=
";H2$:PRINT:N1=PEEK(X+13):GOSUB2260:PRINT"I =";H1$,
 4950 N1=PEEK(X+14):GOSUB2260:PRINT"R =";H1$,:A=PEEK(X+15):V=128:S$="":FORI=7TO0
STEP-1:C$="0":IFINT(A/V)=1THENC$="1":A=A-V
 4955 S$=S$+C$+" ":V=V/2:NEXT:PRINT"F = ";S$:PRINTTAB(24);"⇑├─┼───┼───┼─┼─┤":PRI
NTTAB(24);"S Z   H   P N C":IFFRTHEN606
 4960 GOSUB605:A=M1:GOTO4001
 4999 REM ======= FIN =========
 5000 LIMITMAX:CONSOLEN,C40,S0,24:PRINT"⇓⇓DESASSEMBLEUR est fini.":DEF KEY(1)=RU
N ↵
```

# COPIEZ VOS MASTER DISQUETTES

Vous savez, sans doute, qu'on ne peut faire des copies de l'interpréteur basic, qu'à partir de la disquette mère (MASTER). Une copie de cette disquette devient SUB-MASTER et ne peut plus être recopiée.

Aussi faut-il faire particulièrement attention à ce qu'il n'arrive pas malheur à cette disquette mère.

Pour mettre fin à cette situation désagréable, voici la procédure à suivre :

**1** - Chargez le programme 'UTILITY'.

**2** - Passez sous le contrôle du Monitor en appuyant sur le bouton 'RESET' à l'arrière de l'ordinateur.

**3** - Modifiez l'adresse 13E3 avec l'instruction 'M' du Monitor (remplacez 20 par 18 en 13E3).

**4** - Retournez au programme 'UTILITY' avec l'instruction 'J' à l'adresse 1220.

Ci-dessous, voici ce qui doit apparaître à l'écran :

** MONITOR SB-1510**

* M

M-ADR.$13E3 : modification de l'adresse 13E3

13E3 20 18 : afficher 18 puis 'CR' ou 'ENT'

13E4 23 : appuyer sur 'BREAK' pour arrêter

* J

J-ADR.$1220 : saut à l'adresse 1220

Vous pouvez alors faire des copies de sub-master en utilisant l'instruction 'C' (disquette copy).

**REMARQUE :**

Je n'ai pas testé la validité des autres instructions, après cette modification. Donc si vous voulez initialiser il serait plus prudent de recharger le programme 'UTILITY'.

J.M. SUPOR

# LE MONITEUR SB 1510 PAS A PAS

**Nous commencons dans ce bulletin l'étude du MONITEUR du MZ80-B.**
**Il a trois fonctions essentielles :**
**• ordres moniteur : permet d'afficher ou de modifier les mémoires, et d'enregistrer ou lire sur cassette des zones mémoire.**
**• gestion de l'écran et du clavier, saisie de ligne.**
**• fonctions relatives à l'horloge, et a la « MUSIC ».**

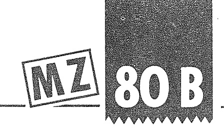
Dans toute la suite, les adresses seront données en hexadecimal, avec les mnemoniques du manuel de référence.
Pour avoir la signification des ordres du Z80, nous vous conseillons l'ouvrage de Alain Pinaud, « Programmer en ASSEMBLEUR », aux éditions du P.S.I. Nous vous rappelons simplement a propos du Z-80 qu'il possède des registres notés A B C D E F H L, le contenu des registres sera noté entre parenthèses, et nous utiliserons le signe correspondant à l'affectation : A 2 ; A est chargé avec la valeur 2, donc A = 2.

Nous étudierons donc le MONITEUR, par adresses croissantes, en trois parties, correspondant aux différentes fonctions.

## LES ORDRES MONITEUR :

**0000 à 0037 : Zone contenant les caractéristiques de fonctionnement - On peut remarquer en particulier :**

| | | |
|---|---|---|
| 00003-0004 | FLPOS : | adresse absolue du curseur dans VRAM. |
| 00005 | ONTYO : | code de durée de la note jouée. |
| 0006-0007 | KEDA et KESTRB : | codes de la dernière touche enfoncée. |
| 000B | SCROST : | numéro de la première ligne d'écran. |
| 000C | SCREND : | numéro de la dernière ligne d'écran. |
| 000D | FLASH : | code ASCII du caractère masqué par le curseur. |
| 000E | FLSDAT : | code ASCII du caractère clignotant affiché (1F ou FLASH). |
| 000F | AMPM : | 0 = matin, 1 = soir. |
| 00013-0014 | SCRST : | adresse le début de VRAM (D000). |
| 0015 | SWRK : | active le BIP de chaque touche si contient 00, pas de sonorisation si FF. |
| 0016-0017 | INIC1 : | heure en secondes. |
| 001B-001C | SCRSIZ : | nombres de caractères disponibles sur l'écran. |
| 001D | TEMPW : | tempo de la musique. |
| 0023-0025 | FOARE : | affichage 00 du pavé numérique. |
| 0026-0027 | KDATW et DATW1 : | codes de la touche enfoncée, ces valeurs sont recopiées dans KEDA et KESTRB si la touche est validée. |
| 002B-002C | SUMDT : | nombre de bits longs, dans un programme, sert de test ; valeur chargée par CKSUM.<br>Utilisés également comme adresse de départ dans les ordres MONITEUR (SHL). |
| 002E-002F | STACK : | rangement de l'adresse du pointeur de pile. |
| 0033-0034 | CSMDT : | nombre de bits longs d'un programme, valeur testée par TVRFY.<br>Utilisés également comme adresse de fin dans les ordres MONITEUR (EHL). |
| 0035 | REPTCT : | temps de répétition des flèches curseur, prend la valeur 40H. |
| 0036-0037 | RATIO : | période de la note jouée. |




**003B à 00AB : Zone d'initialisation des différents modes, et affichage du titre.**

| | | |
|---|---|---|
| 00B1 | ST | Saisie de la commande moniteur (J, M, D, L, S ou V), et branchement à l'adresse correspondante.<br>La ligne introduite est placée dans BUFER (103F ...), seule la première lettre après * est prise en compte.<br>*appelle :* NL PRNT GETL |
| 00F9 | MCLECT | Ordre MONITEUR M (modification) ; modifie les octets mémoire (si CR rien, si BREAK retour en ST)<br>*appelle :* KIN NLPHLS PRTHX PRTS GETLBR 2HEX |
| 0120 | DUMP | Ordre MONITEUR D (visualisation mémoire) ; affichage hexa du contenu des mémoires :<br>adresse de départ : SHL ; adresse de fin : DE ; nombre d'octets affichés par ligne en B (C40 : 8, C80 : 16).<br>*appelle :* SSET ESET NLPHLS PRNTS PRTHX BRKEY |
| 014E | MSAVE | S 2 |
| 01CB | MLOAD | ordres moniteur L : charge MWARK (0216) avec 1 |
| 0217 | MVRFY | V 0 |
| | | puis branchement en MENAME. |
| 0153 | MENAME | Affichage de FILE NAME et saisie du nom du fichier.<br>*appelle :* ?DINT NLMSG GETLBR<br>puis branchement en MLOVE si MWARK < > 2<br>Si ordre S on continue : demande adresse de départ, de fin, de branchement puis écriture sur cassette des caractéristiques à partir de IBUFE (10C0) et des données du programme.<br>*appelle :* SSET ESET KINP HLHEX ?WRI ?WRD |
| 01CF | MLOVE | Chargement des caractéristiques du programme cassette (?RDI), puis comparaison entre le nom saisi et le nom en mémoire (si OD pas de comparaison), si différence : retour.<br>Si MWARK = 0 branchement en MVERY, sinon affichage de LOADING « ... », lecture du programme et branchement a l'adresse d'exécution.<br>*appelle :* ?RDI NAMECK MVERY DSPNAM ?RDD |
| 01F6 | NAMECK | Comparaison des 16 octets d'adresse de début NAME (1061) et des octets suivant BUFER + 10.<br>Si différence, retour à MLOVE ; si égalité (Z=0), retour à l'appel.<br>*appelle :* DSPNAM SAME SERSP MSTOP DEL6 |
| 021A | MVERY | Affichage de VERIFYING « ... », vérification des données cassette ; si erreur, arrêt et retour en ST ; sinon, affichage de OK et retour en ST1<br>*appelle :* DSPNAM ?VRFY NLMSG |
| 024B | JUMP | Ordre moniteur J : branchement à l'adresse HL.<br>*appelle :* KIN |
| 0251 | ?WRI | Préparation à l'écriture du titre sur cassette (la zone IBUFE (10C0 ...) est remplie).<br>D 1, HL IBUFE, BC 80H puis WRI1. |
| 0282 | ?WRD | Préparation à l'écriture des données sur cassette.<br>D 2,HL (DTADR) : adresse de début de lecture, BC (SIZE) : longueur. |
| 025A | WRI1 | Ecriture titre (D = 1) ou données (D = 2) sur cassette. Il y a calcul de la somme de test, démarrage du moteur, affichage du titre si D = 1, puis écriture sur cassette.<br>*appelle :* CKSUM MOTOR DSPNAM TSPE GAP WTAPE |
| 028E | ?RDI | Préparation à la lecture du titre sur cassette.<br>D 4, HL IBUFE, BC 80H puis RD1. |
| 02B2 | ?RDD | Préparation à la lecture des données sur cassette.<br>D 8, HL (DTADR), BC (SIZE) puis RD1. |
| 0297 | RD1 | Lecture titre (D = 4) ou données (D = 8) sur cassette.<br>Il y a démarrage du moteur, lecture de l'identification puis lecture des données.<br>*appelle :* MOTOR TMARK RTAPE SERSP MSTOP |
| 02BE | ?VRFY | Vérification des données sur cassette.<br>D 8 (lecture de données), HL (DTADR) (adresse de début du programme), BC (SIZE) (longueur).<br>Il y a calcul de la somme de test, démarrage du moteur, recherche titre, vérification puis arrêt moteur.<br>*appelle :* CKSUM MOTOR TMARK TVRFY<br>Après ce sous-programme, si erreur C = 1, sinon C = 0. |

| | | |
|---|---|---|
| 02DA | WTAPE | Ecriture sur cassette.<br>E 2 : le programme est écrit deux fois. HL contient l'adresse de départ, BC la longueur.<br>Il y a enregistrement des (BC) octets du programme, de la somme de contrôle (SUMDT), d'un TOP long, de 255 tops court et enfin de l'ensemble précédent (E = 2).<br>*appelle :* WBYTE BRK LONG SHORT |
| 030B | RTAPE | Lecture sur cassette.<br>E 2 : nombre d'enregistrements. HL contient l'adresse de début du programme, BC sa longueur.<br>Il y a attente des données sur cassette, remise a zéro de (SUMDT), lecture du programme, de la somme de contrôle enregistrée et comparaison a (SUMDT) d'où relecture ou affichage d'un message d'erreur.<br>*appelle :* EDGE DLYR RBYTE NLMSG |
| 0358 | TVRFY | Vérification cassette.<br>E 2 : nombre d'enregistrements. HL contient l'adresse de début de programme à vérifier, BC sa longueur.<br>Il y a attente des données sur cassette, comparaison des octets (lus et en mémoire), vérification de (SUMDT).<br>*appelle :* EDGE DLYR RBYTE<br>Après ce sous-programme, retour si OK sinon message d'erreur (sous-programme TAPER 0350). |
| 038 F | WBYTE | Ecriture d'un octet sur cassette.<br>A contient l'octet, B 8 (octet)<br>Enregistrement d'un top long de début d'octet, puis top long ou court suivant la valeur du bit (1 ou 0).<br>*appelle :* LONG SHORT |
| 03A0 | RBYTE | Lecture d'un octet sur cassette.<br>HL 0800. Il y a attente d'un signal, délai de durée comprise entre celle d'un top long et celle d'un top court (41 H), test de signal sur la bande (IN A, (E1H) si rien le top était court d'où C 0, si signal le top était long, d'où, C 1, puis rotation de L a travers C, H est decrémenté, lorsque H = 0 : fin, on charge A avec L.<br>*appelle :* EDGE DLYR |
| 03C7 | GAP | Ecriture de l'identificateur sur cassette (placé en tête de tout enregistrement).<br>D est chargé (1 si titre, 2 si données).<br>Si D = 1, BC 2710, HL 2828 et si D = 2, BC 2 AF8, HL 1414.<br>Il y a alors enregistrement de (BC) tops courts, H longs, L courts et 1 long.<br>*appelle :* SHORT LONG |
| 03F1 | TMARK | Lecture de l'identificateur sur cassette<br>D a la valeur 4 pour le titre, et 8 pour les données.<br>Si D = 4, HL 2828 et si D = 8, HL 1414.<br>Il y a lecture des tops courts, attente des tops longs, comptage de ces tops longs, et retour au top long suivant.<br>*appelle :* EDGE DLYR |
| 0423 | CKSUM | Calcul de la somme de test dans un programme (nombre de bits 1). HL contient l'adresse de départ, BC la longueur.<br>La somme calculée dans DE est placée dans SUMDT et dans CSMDT. |
| 0446 | EDGE | Boucle d'attente de signal sur cassette ou de BREAK.<br>Au retour C = 0 si la lecture est normale, C = 1 si la touche BREAK a été enfoncée. |
| 0457 | MOTOR | Mise en route du moteur, avec les tests de fonctionnement.<br>Il y a blocage du clavier sauf BREAK, vérification cassette enclenchée (sinon message, ouverture de la porte, et attente), passage a PLAY si lecture ( D = 1 ou D = 2), sinon vérification que la cassette n'est pas protégée (sinon message et ouverture de la porte), passage a MOTWG.<br>*appelle :* KBSET NLMSG OPEN BRK DEL1M PLAY MOTWG |
| 048C | OPEN | Ouverture de la porte cassette.<br>*appelle :* DEL1M |
| 0497 | MOTWG | Mise en route du moteur en mode enregistrement. |
| 049B | PLAY | Mise en route du moteur et enregistrement ou lecture suivant le mode choisi.<br>*appelle :* MPLAY FR |
| 04B1 | SERSP | Arrêt du moteur et avance rapide après la lecture.<br>*appelle :* MSTOP DEL6 FFWD |
| 04CE | MSTOP | Arrêt du moteur et passage au mode lecture. |
| 04E9 | FFWD | Avance rapide cassette, et détection éventuelle de BREAK.<br>*appelle :* KBSET BLK3 FR |
| 0500 | DELT | Attente 4s. BC 0F2B puis D1M |
| 0511 | DEL6 | Attente 300us. BC 0123 puis D1M |
| 0517 | DEL1M | Attente 2s. BC 0796 puis D1M |
| 0504 | D1M | Temporisation suivant la valeur de (BC). |

| | | |
|---|---|---|
| 051D | SHORT | Enregistrement d'un top court sur cassette.<br>Sortie d'un signal pendant une durée correspondant a un délai de 2AH, puis rien pendant 25H.<br>*appelle :* DLY |
| 0539 | LONG | Enregistrement d'un top long sur cassette.<br>Sortie du signal pendant une durée correspondant a un délai de 5AH, puis rien pendant 55H.<br>*appelle :* DLY |
| 0554 | DLYR | Attente pour la lecture de cassette. A 41 H puis DLY.<br>Cette valeur 41H est comprise entre 2AH et 5AH (tops court et long). |
| 055B | DLY | Boucle d'attente. La valeur est chargée en 055C.<br>Permet d'avoir des temporisations courtes pour les tops d'enregistrement |
| 0562 | BRKEY | Bloque le clavier et détection de la touche BREAK.<br>*appelle :* KBSET |
| 056C | BRK | Détecte BREAK. Si BREAK, C = 1 sinon C = 0. |
| 0571 | KBSET | Bloque le clavier, prépare à la lecture de BREAK. |
| 057A | SSET | Affiche S-ADR$, saisie d'une adresse en hexa.<br>Cette adresse est placée dans SHL (002B-002C).<br>*appelle :* KIN |
| 0583 | ESET | Affiche E-ADR$, saisie d'une adresse en hexa.<br>Cette adresse est placée dans EHL (0033-0034).<br>*appelle :* KIN |
| 058C | KIN | Saisie de 4 chiffres hexa sous forme ASCII, transformés en nombre place dans HL.<br>*appelle :* KINP HLHEX |
| 0598 | KINP | Saisie de 4 chiffres hexa (code ASCII).<br>Il y a affichage de S ou E - ADR$ dans BUFER (103F ..) saisie de ligne, puis comparaison de (BUFER + 7) à OD.<br>*appelle :* NLMSG GETLBR |
| 05B6 | NLMSG | Affichage nouvelle ligne et de la chaîne commençant à l'adresse contenue dans DE, jusqu'à rencontrer OD.<br>*appelle :* NL MSGX |
| 5BC | NLPHLS | Affichage nouvelle ligne, du contenu de HL puis d'un espace.<br>appelle : NL PRTHL PRNTS |
| 05C5 | GETLBR | Saisie d'une ligne ou de BREAK. Retour en ST (00B1) si BREAK.<br>*appelle :* GETL |
| 05CF | DISPNAM | Affichage d'un message (adresse de début : (DE), puis du nom du programme (adresse de début : (NAME)).<br>*appelle :* NLMSG MSGX |
| 05D8 | PRTHL | Affichage du contenu de HL.<br>(HL), nombre hexa, est affiché sous forme code ASCII.<br>*appelle :* PRTHX |
| 05DD | PRTHX | Affichage du contenu de A (nombre hexa) sous forme de code ASCII.<br>On transforme d'abord les 4 bits supérieurs puis les 4 bits inférieurs.<br>*appelle :* ASC PRNT |
| 05F3 | ASC | Convertit les 4 bits inférieurs de A (chiffre hexa) en code ASCII.<br>L'opération est la suivante : A A + 30 H, si < 3AH, A A sinon A A + 07H. |
| 05FD | HEX | Convertit A (code ASCII) en chiffre hexa sur les 4 bits inférieurs.<br>L'opération est la suivante : si A> = 47 H, erreur (C 1) sinon A A – 30 H. |
| 0614 | HLHEX | Les 4 codes ASCII placés en DE, DE + 1, DE + 2, DE + 3 sont convertis en un nombre hexa placé dans HL.<br>*appelle :* 2HEX |
| 0623 | 2HEX | Convertit 2 codes ASCII placés en DE et DE + 1 en valeur hexa mise dans A.<br>Les 4 bits supérieurs de A (DE), les 4 bits inférieurs (DE + 1). Si erreur : C 1.<br>*appelle :* HEX |
| 063A | SAME | Comparaison de deux blocs de mémoire.<br>Adresse de début premier bloc : (DE)<br>Adresse de début deuxième bloc : (HL)<br>Longueur à comparer : (B)<br>Il y a arrêt si on trouve OD dans la chaîne DE, ou si toute la chaîne a été parcourue. Si différence Z 1, sinon Z 0. |
| 064F | ?CLER | Remise a zéro d'une partie de la mémoire (A 0). |
| 0650 | ?DINT | Chargement d'un bloc mémoire avec la valeur de A.<br>Adresse de début : (HL), longueur : (B). |
| 0655-0641 | | Différents messages pouvant être affichés. |

A suivre...

Jean Millet




## JEUX

| | | |
|---|---|---|
| D001:ROLLAND GAR | Jeu de tennis a 2 joueurs jouant ensemble | 2 Ko. |
| D002:LABYRINTHE | Il faut sortir du labyrinthe le plus vite possible | 2 Ko. |
| D003:CASSE BRIQU | Un classique. Detruisez le mur le plus rapidement possible | 2 Ko. |
| D005:BIO ASTRAL | Determiner votre biorythme astral a partir des 3 courbes sur CE 150+commentaires | 8 Ko. |
| D006:ESPER. VIE | Calculez votre esperance de vie statisque | 2 Ko. |
| D007:JACK POT | Retrouvez les joies de LAS VEGAS sur votre PC 1500 (les deceptions aussi !!) | 2 Ko. |
| D008:D & D 1 | Simulation de Donjon et Dragon | 2 Ko. |
| D009: 421 | Jouez au 421 comme au bistrot | 8 Ko. |
| D011:JEU ROYAUME | Gerez votre royaume avec ses habitants et ses richesses minieres | 2 Ko. |
| D012:COMBAT DUEL | battez vous en duel contre le PC 1500 ou contre un autre joueur | 4 Ko. |
| D013:LOGIC-LETTR | Jeu de logique pour reconstituer un damier compose de lettres | 2 Ko. |
| D015:D & D 2 | Vous devez atteindre la salle au tresor en evitant les dangers | 2 Ko. |
| D016:SNAKE | Conduisez un serpent qui s'allonge de plus en plus en lui faisant manger des oeufs | 4 Ko. |
| D017:D & D 3 | Promenez vous dans les piece d'un chateau pour y decouvrir un tresor) GRAPHIQUE | 8 Ko. |
| D018:SIGNES CACH | Un jeu de memoire visuelle ou il vous faut reconstituer des paires | 4 Ko. |
| D019:HANOI | Celebre casse tete | 4 Ko. |
| D020:TCHOUKA | Casse tete plus sa resolution par l'ordinateur si vous abandonnez !! | 4 Ko. |
| D021:LABY GRAPH | Essayez de trouver la sortie du labyrinthe. GRAPHIQUE | 2 Ko. |
| D022:TRAIN | Dessin anime decrivant l'arrivee en gare d'un train du far-west | 4 Ko. |
| D023:MORPION | Jouez au morpion 10X10 . Tres fort. trace sur CE 150 si elle est presente | 8 Ko. |
| D024:BALOMUR | Comprend 2 phases de jeu | 2 Ko. |
| D025:PENDU 1 | Un nombre impressionnat de mot en memoire pour ce pendu | 16 Ko. |
| D026:ROSE.VENTS | Trace sur la CE 150 une rose des vent | 2 Ko. |
| D027:TRESOR | Jeu de role. Retrouvez un tresor sur une ile. Un des meilleurs jeu de ce type | 16 Ko. |
| D028:GRAPHIQUES | Trace sur la CE 150 des graphismes suivant un coefficient que vous rentrez | 2 Ko. |
| D029:TRAJECTOIRE | Etude sous forme ludique de la trajestoire d'un corps CE 150 | 4 Ko. |
| D030:PSY | Tenez une conversation en Anglais avec votre PC 1500 qui vous repondra | 8 Ko. |
| D031:NU | Dessin d'un nu sur CE 150 ou WATANABE WX 4671 | 8 Ko. |
| D032:SAVON ZIP.. | Essayez de rattraper un savon au fond de votre baignoire ... | 2 Ko. |
| D033:CHOUETTE | Dessin sur la CE 150 d'une tete de chouette | 4 Ko. |
| D034:POLYGONE | Dessin sur la CE 150 d'un polygone decore | 2 Ko. |
| D035:MACARON | Dessin sur la CE 150 d'un superbe macaron | 2 Ko. |
| D036:PAON | Dessin sur la CE 150 d'un motif rappelant la queue du paon | 2 Ko. |
| D037:BOULE | Dessin sur la CE 150 d'une boule tres decorative | 2 Ko. |
| D038:CROIX | Dessin sur la CE 150 d'une croix tres decorative | 2 Ko. |
| D039:SYMETRIE | Dessin de surface decoree sur CE 150 | 2 Ko. |
| D040:COURBES DIF | Trace artistique de courbes differentielles | 2 Ko. |
| D041:COURSE | Simulation d'une course de voiture sur votre ecran | 2 Ko. |
| D042:OBELIX | Dessin a la CE 150 du celebre OBELIX | 4 Ko. |
| D043:SNOOPY | Dessin de SNOOPY sur la CE 150 | 4 Ko. |
| D044:EXO7 | Echappez au missile qui vous poursuit . Sur CE 150 | 2 Ko. |
| D045:FRANCE | Dessin sur la CE 150 de la carte de FRANCE | 4 Ko. |
| D046:MAPPEMONDE | Dessin sur la CE 150 d'une mappemonde | 8 Ko. |
| D047:DEMINEUR | Deminez votre ecran pour faire passer un paquebot | 8 Ko. |
| D048:TEMPTER | Circulez dans un labyrinthe qui est visualise en 3D pour rechercher un tresor | 4 Ko. |
| D049:LOCO | Conduisez votre locomotive au terminus en achetant l'eau et le charbon en route | 4 Ko. |
| D050:MZ 80B | Dessin sur la CE 150 du grand frere du PC 1500 | 4 Ko. |
| D051:SHARP | Dessinez sur la CE 150 le celebre nom ! | 4 Ko. |
| D052:ASTERIX | Dessin sur la CE 150 du buste d'ASTERIX | 4 Ko. |
| D053:PC 1500 | Dessin sur la CE 150 de moi meme !!! | 4 Ko. |
| D054:PICSOU | Dessin sur la CE 150 d'on superbe PICSOU | 8 Ko. |
| D055:TBN | Jeu d'aventure avec mode d'emploi en Japonais. Avis aux amateurs !! | 8 Ko. |
| D056:GOLF | Jouez une partie de golf sur votre CE 150. Tres belle simulation | 8 Ko. |
| D057:ASTEROIDES | Jeu entierement en langage machine sur votre ecran | 4 Ko. |
| D058:SER.OIS | Jeu en L M. Compatible paddle. Entourez la mouche avec le serpent . 2 joueurs | 4 Ko. |
| D059:KABOOM | Evitez que les meches arrivent a l'explosif | 4 Ko. |
| D060:STEEPLE CH. | Faites passer les obstacles a votre fidele coursier | 4 Ko. |
| D061:SMASHER | Essayez de faire traverser l'ecran a votre point sans vous faire ecraser | 4 Ko. |
| D063:QUICKEYE | Retrouvez un maximun de chiffres sur ceux affiches au debut | 4 Ko. |
| D064:UFO | Detruisez les envahisseurs qui infestent votre ecran | 4 Ko. |
| D065:CANON | Detruisez le plus possible de soucoupes qui appairaissent a l'ecran | 4 Ko. |
| D066:FIRE MAN | Essayer d'eteindre un feu avant qu'il ne vous brule | 4 Ko. |
| D068:CAL METEO | Calculez pour les mois a venir le temps qu'il va faire !!! | 8 Ko. |
| D069:MANAGER | Gerer vos entreprises | 8 Ko. |
| D070:CHIF. CASES | Se joue a deux joueurs. Jeu de reflexion | 2 Ko. |
| D071:PENDU | Jeu celebre sur votre affichage | 4 Ko. |
| D072:CMPT TAUPE | Jeu pour les petits enfants. Retrouver un nombre de taupes affichees | 4 Ko. |
| D073:TIR AU VOL | Essayer d'ajuster votre tir pour detruire un soucoupe | 4 Ko. |
| D074:LETTRIX | Reconstituez un mot defini a l'avance | 2 Ko. |
| D075:CHOPLIFTER | Jeu graphique. Detruisez vos ennemis et recuperer vos amis au sol | 4 Ko. |
| D076:PENDU CHRON | Une autre version de ce jeu. Le temps est chronometre | 4 Ko. |
| D077:DECATHLON | 10 epreuves pour tester votre forme. Epuisant !!! | 8 Ko. |
| D078:RUNNER | Une course de 100 M tres graphique | 4 Ko. |
| D079:TARGET | Apprenez a tirer a l'arc sur votre CE 150 | 4 Ko. |
| D080:TANK | Vous | 2 Ko. |
| D081:TREASURE IS | Retrouvez un tresor dans l'ile. Pas de mode d'emploi !! | 8 Ko. |
| E001:STILL NACHT | Melodie | 2 Ko. |
| E002:CHANSON | Melodie | 2 Ko. |
| E003:FRERE JAC | Melodie | 2 Ko. |
| E005:TONALITE | Calcul d'une tonalite avec trace de la gamme | 4 Ko. |
| E006:TRANSPOCLE | Transposition d'une cle en une autre avec trace de la gamme | 8 Ko. |
| E007:PC BLUES | Melodies en LM. SUPERBE !! | 8 Ko. |
| E008:PRG ECRT MU | Programme d'ecriture musicale | 4 Ko. |

## UTILITAIRES

| | | |
|---|---|---|
| B001:TYPE WRITER | Mini traitement de texte sur CE 150 multidirectionnel et en couleur | 2 Ko. |
| B002:DATEUR | Imprime la date sur vos listings | 2 Ko. |
| B003:CALPE | Edite un calendrier de l'annee de votre choix | 2 Ko. |
| B004:OPECAL | Effectue des operations sur les dates | 2 Ko. |
| B006:LETTRES REL | Ecriture en relief sur la CE 150. Tres beau | 2 Ko. |
| B007:ORDIROUTE | Tout sur votre parcours de voiture | 4 Ko. |
| B008:SAVERAM | Permet de stocker des programmes en RAM et de les recuperer ensuite | 8 Ko. |
| B009:COMPTAROT | Permet de compter les points au tarot | 4 Ko. |
| B010:COMPTENNIS | Permet de compter les points au tennis | 4 Ko. |
| B011:GESCASSETTE | Gestion d'un fichier pour vos programmes sur cassette | 4 Ko. |
| B013:LGN BRISEES | Trace d'un graphe par le systeme des lignes brisees | 2 Ko. |
| B014:STAT LOTO | Traitement statistique des resultats du loto | 4 Ko. |
| B015:FICHES TEL | Gestion d'un fichier telephonique | 4 Ko. |
| B016:GFICHES | Gestion de fichier parametrable | 8 Ko. |
| B017:EDIT ADRES | Edition sur la CE 150 d'une etiquette a coller sur l'enveloppe | 2 Ko. |
| B018:SUPERDRIVER | Nouvelle gestion du clavier -REPEAT BEEP ANSWER - LM | 2 Ko. |
| B019:PLM | Editeur de langage machine en Z 80 | 8 Ko. |
| B020:DESSIN | Utilitaire d'aide au dessin sur l'ecran-Possibilite de dessins animes | 8 Ko. |
| B021:INKEY# | Routine LM creant un super INKEY$ acceptant plusieurs touches ensemble | 2 Ko. |
| B022:BIDI LIST | Listing bidirectionnel d'un programme sur CE 150 en 80 colonnes LM | 2 Ko. |
| B023:SCROLLINGS | 4 routines LM faisant les 4 types de scrolling | 2 Ko. |
| B024:ROULEAUX | 4 routines LM faisant defiler l'ecran en rouleau | 2 Ko. |
| B025:UTILAFF | 4 routines LM destinees a ameliorer les possibilites d'affichage | 2 Ko. |
| B027:MAD | Moniteur Assembleur Desassembleur en mnemoniques officiels | 8 Ko. |
| B029:CODEX | Codage et decodage de messages d'apres une cle | 2 Ko. |
| B030:DES CE 158 | Desassembleur en mnemoniques officiels sur deux colonnes avec CE 158 | 8 Ko. |
| B031:BASIC 1501 | 33 nouvelles fonctions BASIC pour votre PC 1500 | 8 Ko. |
| B032:ETIQUETTES | Copie d'ecran parametree sur CE 150 | 2 Ko. |
| B033:BASIQUOIS | Redefinition des fonctions BASIC en Francais | 8 Ko. |
| B034:BASIC ETEND | 3 nouvelles fonctions BASIC pour votre PC | 4 Ko. |
| B035:GRAPH 3D | Definition d'un objet en 3D et representation sur CE 150 | 4 Ko. |
| B036:MONITEUR | Moniteur hexadecimal servant a rentrer les programmes du bulletin | 2 Ko. |
| B037:GESTION ERR | Nouvelle gestion des erreurs | 4 Ko. |
| B038:CHECKSUM | Permet de recuperer un programme apres un NEW ou ERROR 44 | 2 Ko. |
| B040:DEMERGE | Permet de supprimer l'effet d'un MERGE | 2 Ko. |
| B041:DUMP ROT1 | Effectue un DUMP en ROTATE 1 ce qui augmente la lisibilite CE 150 | 2 Ko. |
| B042:GRAPHOCOLOR | Presentation de pages sur MINITEL de preference couleur | 16 Ko. |
| B043:TITRE K7 | Edition d'etiquettes a mettre sur vos cassettes pour voir leurs contenus | 4 Ko. |

## ENSEIGNEMENT – SCIENTIFIQUES

| | | |
|---|---|---|
| A002:COMPLEXES | Effectue toutes les operations sur les complexes | 8 Ko. |
| A003:RACINE POL | Calcule la racine d'un polynome | 2 Ko. |
| A004:BALISTIQUE | Calcule et trace la trajectoire d'un projectile CE 150 | 2 Ko. |
| A005:TRIANGLE | Calcule et trace un triangle defini par trois parametres CE 150 | 2 Ko. |
| A006:HYPERBOLIQ | Donne acces aux fonctions hyperboliques | 2 Ko. |
| A007:CORRELATION | Calcule diverses correlations (logarythmique | 4 Ko. |
| A008:RES.EQUATIO | Resolution des equations algebriques | 2 Ko. |
| A009:CONV.COORD | Conversion de coordonnees d'un repere a un autre | 2 Ko. |
| A010:SCIENTIBAS | Donne acces a de nouvelles fonctions scientifiques BASIC LM | 2 Ko. |
| A011:NB.PREMIERS | Petit utilitaire calculant les nombres premiers | 2 Ko. |
| A012:COMP.FNC | Autoprogrammateur acceptant toutes les fonctions | 2 Ko. |
| A013:HEX.DEC | Traduction de l'hexadecimal en decimal a l'interieur d'un programme | 2 Ko. |
| A014:FACT.NOT.SC | Calcul d'une factorielle en notation scientifique | 2 Ko. |
| A015:FACT.PRECIS | Calcul d'une factorielle avec tous les chiffres | 2 Ko. |
| A016:ZERO FNC | Calcule le zero d'une fonction | 2 Ko. |
| A017:HP 41 | Simulation d'un HP 41 en mode calcul (piles et nombreuses fonctions ) | 2 Ko. |
| A018:POLYNOME | Traitement de polynome sous forme algebrique | 8 Ko. |
| A019:ARITHMETIQ | Traitement arithmetique de donnees | 2 Ko. |
| A020:DERIVATEUR | Calcule la derivee en litteral d'une expression algebrique | 8 Ko. |
| A021:SYRACUSE | Calcule la distance d'un nombre pour le probleme de Syracuse LM | 2 Ko. |

## DIVERS

| | | |
|---|---|---|
| C001:ANIM. VENTE | Programme d'animation des ventes | 4 Ko. |
| C002:RENT. INVES | Calcul de la rentabilite d'un investissement | 4 Ko. |
| C003:PLAN EPARG | Calcul du revenu d'un plan d'epargne | 4 Ko. |
| C004:INTERET COM | Calcul des interets composes | 4 Ko. |
| C005:AMORT.EMP | Calcul de l'amortissement d'un emprunt | 4 Ko. |
| C006:COMPTA ANA | Gestion de comptabilite analytique avec graphismes | 8 Ko. |
| C007:RAP.BANCAIR | Verification des comptes bancaires | 4 Ko. |
| C008:SALLE VENTE | Gestion d'une salle des ventes | 8 Ko. |
| C010:POLARIS.AMP | Etude de la polarisation d'un ampli | 4 Ko. |
| C011:TAB.VERITE | Etude d'une table de verite en electronique | 2 Ko. |
| C012:ACCI.TRAVAI | Calcul du cout des accidents du travail | 2 Ko. |
| C013:GES.CABMED | Gestion d'un cabinet medical | 4 Ko. |
| C014:ANA.PARETO | Analyse de PARETO + trace sur CE 150 | 4 Ko. |
| C016:DRAGAGE | Calcul de terrassement par dragage + trace sur CE 150 | 8 Ko. |
| C020:EPHEMERIDES | Calcul des ephemerides des planetes et etoiles | 8 Ko. |
| C021:CALCSHARP | Reprend le systeme des CALC + trace | 8 Ko. |
| C022:VENTILATION | Ventilation des ventes | 8 Ko. |
| C023:CAL.FINANC. | Effectue toutes sortes de calcul sur les emprunts | 8 Ko. |
| C024:NAVEASTRO | Calculs pour la navigation astronomique | 8 Ko. |
| C025:OSCIL.AMORT | Calcul et trace de l'amortissement d'un oscillateur | 4 Ko. |
| C026:RLC SERIE | Calcul d'un circuit RLC en serie et trace | 4 Ko. |
| C027:JUPITER | Position des satellites de Jupiter a un moment precis + trace | 4 Ko. |
| C028:FOKKER27500 | Calcul du carburant et du temps de vol sur FOKKER 27/500 | 8 Ko. |

# BULLETIN D'INSCRIPTION AU CLUB DES SHARPENTIERS

☐ Je m'inscris au CLUB DES SHARPENTIERS

Je bénéficie de tous les avantages du CLUB

Je suis abonné pour 1 AN au BULLETIN du CLUB

Je vous joins mon règlement

☐ FRANCE : 160 F

☐ ETRANGER : 200 F

CHEQUE N° ............. BANQUE ......

DATE SIGNATURE

n°11

NOM .................... PRÉNOM ..................

ADRESSE ..............................................

CODE POSTAL / VILLE ..................................

PAYS ..................................................

PROFESSION ................... ÂGE .................

MACHINE POSSEDÉE ................. DEPUIS ...............

ACHETÉE CHEZ ...........................................

UTILISATION PRINCIPALE DE VOTRE MACHINE ....................

..........................................................

Club des Sharpentiers 151/153, avenue Jean-Jaurès 93307 AUBERVILLIERS CEDEX Tél : 834.93.44